



# ESCUELA TÉCNICA SUPERIOR DE INGENIEROS INDUSTRIALES Y DE TELECOMUNICACIÓN

Titulación:

INGENIERO EN INFORMÁTICA

Título del proyecto:

USO DE ONTOLOGÍAS PARA LA DISTRIBUCIÓN DEL TRÁFICO EN  
SITUACIONES DE EMERGENCIA

Antonio Jesús Bermejo Morales

Jesús Villadangos Alonso

Pamplona, 27 de Julio de 2012

## TABLA DE CONTENIDOS

CAPÍTULO 1: INTRODUCCIÓN .....	5
PRESENTACIÓN DEL PROYECTO .....	6
MARCO QUE ENGLOBA EL PROYECTO .....	8
CONTRIBUCIONES .....	9
ESTADO DEL ARTE .....	11
CAPÍTULO 2: HERAMIENTAS .....	15
CAPÍTULO 3: REQUISITOS .....	18
REQUISITOS FUNCIONALES: .....	19
REQUISITOS TÉCNICOS .....	20
REQUISITOS TECNOLÓGICOS .....	21
CASOS DE USO .....	21
CAPÍTULO 4: ANÁLISIS .....	22
ANÁLISIS DE LA SITUACIÓN .....	23
ONTOLOGÍA .....	24
SIMULADOR - LENGUAJE DE PROGRAMACIÓN .....	25
CAPÍTULO 5: DISEÑO .....	28
ARQUITECTURA DEL SISTEMA .....	29
a. APLICADA AL ESCENARIO REAL .....	29
b. APLICADA AL SIMULADOR .....	30
CLASES .....	31
CAPÍTULO 6: METODOLOGÍA .....	34
CAPÍTULO 7: IMPLEMENTACIÓN .....	38
ONTOLOGÍA, PRIMERA VERSIÓN .....	40
ONTOLOGÍA, VERSIÓN ACTUAL .....	41
SIMULADOR, PRIMERA VERSIÓN .....	45
SIMULADOR, VERSIÓN ACTUAL .....	48
CAPÍTULO 8: PRUEBAS .....	52
CAJA NEGRA .....	53
CAJA BLANCA .....	54
PRUEBAS DE CARGA .....	56
CAPÍTULO 9: DISCUSIÓN .....	58
CAPÍTULO 10: CONCLUSIONES .....	61
BENEFICIOS DEL PROYECTO Y SUS CARACTERÍSTICAS .....	62
LIMITACIONES DEL PROYECTO Y SUS CARACTERÍSTICAS .....	63
CAPÍTULO 11: LÍNEAS FUTURAS .....	65
LOGROS CONSEGUIDOS .....	68
AGRADECIMIENTOS .....	69
REFERENCIAS .....	70
APÉNDICE A .....	71
APÉNDICE B .....	72
APÉNDICE C .....	75
APÉNDICE D .....	77
APÉNDICE E .....	79

# TABLA DE FIGURAS

## ILUSTRACIONES

Ilustración 1: Casos de uso (Car2Car Communication Consortium). .....	12
Ilustración 2: Efecto de situaciones dispersas y densas sobre patrones de movilidad.....	13
Ilustración 3: Protégé. ....	16
Ilustración 4: Java Platform. ....	16
Ilustración 5: Diagrama de secuencia - Simulador. ....	26
Ilustración 6: Diagrama de secuencia - Escenario real. ....	27
Ilustración 7: Primeros pasos en la comunicación. ....	29
Ilustración 8: Proceso en el interior del vehículo. ....	30
Ilustración 9: Progreso conseguido. ....	30
Ilustración 10: Arquitectura aplicada al simulador. ....	31
Ilustración 11: Clases que componen el simulador. ....	32
Ilustración 12: Propiedades de datos de la ontología.....	41
Ilustración 13: Jerarquía de clases (ver Apéndice A). ....	42
Ilustración 14: Consulta SPARQL para la inserción de un vehículo.....	46
Ilustración 15: Ejecución de la consulta SPARQL. ....	47
Ilustración 16: Extraer acciones.....	47
Ilustración 17: Estructura del simulador. ....	48
Ilustración 18: Vehículo posicionado sobre la carretera.....	48
Ilustración 19: Simulador desarrollado.....	49
Ilustración 21: Inferencia con Jess.....	50
Ilustración 20: Nueva manera de insertar individuales.....	49
Ilustración 22: Caja negra. ....	53
Ilustración 23: Extracto de código que asigna huecos adelantados.....	55
Ilustración 24: Prueba de caja blanca. ....	56
Ilustración 25: Prueba de caja blanca (simplificada). ....	56
Ilustración 26: Vehículo conducido sin interacción humana.....	59
Ilustración 27: Ontología resultante a partir de la extensión de la ontología A3ME [16].....	66
Ilustración 28: Jerarquía de clases de la ontología. ....	71
Ilustración 29: Nivel 0, Traffic. ....	72
Ilustración 30: Nivel 1, Simulador.....	72
Ilustración 31: Nivel 1.1, Posicionar vehículo.....	72
Ilustración 32: Nivel 1.2, Establecer velocidad. ....	73
Ilustración 33: Nivel 1.3, Start.....	73
Ilustración 34: Nivel 1.3.1, No más vehículos.....	73
Ilustración 35: Nivel 1.3.2, Utilizar ontología.....	74
Ilustración 36: Nivel 1.3.3, Mover vehículos.....	74
Ilustración 37: Diagrama de Colaboración - Simulador. ....	75
Ilustración 38: Diagrama de Colaboración - Escenario real. ....	76
Ilustración 39: Reglas SWRL. ....	78

## TABLAS

Tabla 1: Descripción detallada del trabajo realizado y comparación con otros. ....	10
Tabla 2: Comparación general con otros trabajos realizados. ....	14
Tabla 3: Propiedades de datos. ....	41
Tabla 4: Clases de la ontología. ....	42
Tabla 5: Escenarios de emergencia, y correspondientes sugerencias por parte de la ontología. Vehículos numerados del 1 al 7 (parte izquierda); y sugerencias de la ontología (parte derecha). ....	43
Tabla 6: Elementos de la ontología. ....	44
Tabla 7: Comparativa de tiempos obtenidos al inferir. ....	50
Tabla 8: Comparativa de tiempos obtenidos al inferir. ....	51
Tabla 9: Prueba de caja negra. ....	54
Tabla 10: Comparativa de tiempos obtenidos al inferir. ....	57

## **CAPÍTULO 1: INTRODUCCIÓN**

En este capítulo se aporta una presentación del proyecto realizado y se sitúa a su vez dentro del marco que lo engloba. Además, se presentan una serie de contribuciones aportadas mediante su realización, y se explica detalladamente estado del arte, es decir, la situación actual sobre todo lo desarrollado hasta el momento, y que se está actualmente desarrollando.

## **PRESENTACIÓN DEL PROYECTO**

Hoy en día la sociedad tiene muchas preocupaciones en su vida diaria sobre diferentes ámbitos como la economía, el trabajo, la política, las guerras, la pobreza, la seguridad,... En el presente proyecto nos vamos a centrar en el último de ellos, y más concretamente en la seguridad en la carretera.

Los Gobiernos y las empresas están muy concienciados con ello e invierten gran parte de sus presupuestos en tratar de concienciar a la sociedad, y a su vez combatir esta gran lacra del mundo moderno, superpoblado de vehículos.

Las mejoras tecnológicas se han ido incorporando poco a poco, pero todavía hace falta un arduo trabajo por delante para evitar que se suceda el gran número de víctimas ligadas a la carretera.

Varias situaciones en las cuales la seguridad de los usuarios de la carretera se ve en peligro son las siguientes:

- Objeto en la carretera.
- Condiciones atmosféricas adversas.
- La dificultad intrínseca de la carretera (trazado, pavimento,...).
- Vehículo en emergencia que necesita paso libre.

En este proyecto todo el trabajo se centra en el último caso, en el que un vehículo en emergencia necesita que se le deje paso libre de la manera más rápida posible, para que pueda realizar su trabajo con diligencia.

Hay una importante necesidad de un estudio profundo sobre esta situación en concreto, y es por lo que se ha elegido de entre muchas otras. Se entiende que se deben buscar soluciones efectivas que ayuden en gran medida a la distribución del tráfico de una manera eficiente.

Hasta el momento no hay noticias de haberse llegado a algún punto de éxito en este campo, con las mínimas garantías. Ello se deduce de los estudios realizados y también del hecho de que no hay ningún vehículo en la actualidad que incorpore tecnologías que solucionen el problema expuesto en este documento.

Por lo tanto, lo que se propone en este proyecto es una solución a este importantísimo tema relacionado con la seguridad en la carretera en situaciones de emergencia.

Hoy en día, los vehículos que circulan por las carreteras son elementos totalmente individuales unos de otros, no existe ningún tipo de comunicación entre ellos en la cual se intercambien información. Por ello, lo que se propone no es solamente el establecimiento de comunicaciones, sino también una coordinación, de manera que los datos intercambiados entre ellos tengan una funcionalidad y se utilicen para un fin concreto y común: la solución de la situación de emergencia.

Para ello, es necesario crear una red de intercomunicación por la cual se transmita toda la información. Esta red será una VANET (Vehicular Ad-hoc NETwork), basada en la comunicación inalámbrica, sin depender de un punto central por el cual pase toda la información (los nodos se comunican directamente entre sí, comunicación distribuida).

Sin embargo, el proyecto no se centra en las comunicaciones dentro de la red, sino que obvia lo puramente técnico o tecnológico para estudiar específicamente todo lo relacionado con estas situaciones de emergencia en las que hay que dejar paso libre a un vehículo prioritario.

Todo esto se entiende, por supuesto, manteniendo la seguridad de todo lo que le rodea y se ve afectado por la situación. La rapidez de reacción está unida con la seguridad en este proyecto.

Cada vez que se produce una situación extraordinaria en la carretera, aparecen peligros y situaciones complicadas para los conductores involucrados. Éstos tienen que utilizar todas sus habilidades para encararlas y poder actuar en consecuencia, tratando de actuar de la manera correcta.

Sin embargo, en la mayoría de los casos, los conductores inmersos en este tipo de situaciones no saben cómo reaccionar o lo hacen de manera incorrecta.

El escenario a estudiar ya tiene una dificultad inherente, porque ya hay una situación de emergencia a solucionar. Si a esto se le añaden nuevas situaciones de peligro derivadas del comportamiento de los conductores, se puede llegar a generar nuevos problemas a afrontar, e incluso nuevos accidentes. Esto significa que hay que tener especial atención y cuidado en cómo se organizan las reacciones a estas situaciones de peligro.

Varias de estas nuevas dificultades generadas, derivadas de la propia situación de emergencia, podrían ser las siguientes:

- El servicio que se espera que realice el vehículo en emergencia sería retrasado y, consecuentemente, la vida de otras personas que necesitan de su servicio podría correr serio peligro.
- Habría una nueva situación de peligro, la cual afectaría a nuevos usuarios, así como a los que ya estaban afectados por la inicial. De esta manera los conductores de los vehículos tendrán que tener especial cuidado con sus comportamientos mientras se sucede la situación de peligro.
- Nuevos accidentes podrían sucederse, transformándose una simple situación extraordinaria en otra mucho más drástica y peligrosa.

Por lo tanto, se puede deducir la imperiosa necesidad de estudiar este escenario y proponer una serie de ideas que puedan ayudar en este tipo de situaciones de incertidumbre, de manera que se produzcan menos a menudo, e incluso que desaparezcan.

De esta manera, lo que se persigue como objeto del proyecto es una distribución del tráfico, de manera segura e intentando modificarlo en la menor medida posible. Se sugerirá una solución personalizada a cada conductor que esté afectado en el escenario, y así se disminuirá en gran medida la posible incertidumbre que le pudiera surgir, consiguiendo un doble objetivo: dejar paso libre al vehículo en situación de emergencia, y además hacerlo de una manera segura.

Para ello, serán necesarias y primordiales las comunicaciones establecidas entre los diferentes vehículos. Esas comunicaciones se realizarán a través de la ya comentada red VANET, compuesta por todos los vehículos

Esto se realizará extrayendo datos concretos de la red y añadiéndolos a un sistema de toma de decisiones, llamado ontología, previamente diseñado y creado para este propósito. Esta ontología propondrá al conductor una serie de soluciones acordes a la situación dada, el cual será el responsable de llevarlas a cabo o no.

Los datos que servirán de base para esa toma de decisiones serán los que definen el movimiento de cada vehículo: su posición y velocidad.

La parte que concierne a este proyecto en concreto, el estudio y desarrollo de un sistema de toma de decisiones, forma parte de un proyecto más amplio. Dicho proyecto está siendo financiado por el Gobierno Foral de Navarra mediante una Beca de Tecnólogos, y engloba todo el desarrollo necesario para la implantación total del sistema en un vehículo: desde los protocolos a utilizar en las comunicaciones, el medio físico por el que enviar los datos, los dispositivos necesarios, el tamaño de paquete a enviar, la formación de la red, el control y organización de las comunicaciones, hasta la manera de mostrar al conductor del vehículo la solución encontrada.

Está planificado que el resultado final del mismo sea una instalación completa, la cual será integrada en vehículos. Dichos vehículos serán los taxis de una ciudad concreta, los cuales instalarán los dispositivos y comprobarán los beneficios obtenidos de la utilización de este sistema desarrollado.

## **MARCO QUE ENGLOBA EL PROYECTO**

Tal y como se explica en [18], se desea que los vehículos, basándose en sus propias mediciones y en la información proveniente tanto de otros vehículos cercanos como de infraestructuras, sean capaces de cooperar para anticiparse, detectar y evitar situaciones peligrosas e indeseadas en un espacio y tiempo reducidos.

Hay cantidad de proyectos enumerados en el artículo anteriormente mencionado (AKTIV, C2C-CC, CityMobil, COM2REACT, COOPERS, CVIS, CyberCars2 and CyberMove) que están trabajando o han trabajado en la seguridad en el transporte. El proyecto actual está englobado dentro de un marco que no solamente basa su desarrollo en la seguridad, sino también en la eficiencia en el transporte y en servicios para los usuarios.

Después de realizar un estudio profundo sobre el tema, en dicho artículo se establece que los vehículos deberían proveer funcionalidades de entorno de red, interfaces para los procesadores y sensores, tareas de computación para las aplicaciones y también una interfaz de usuario. Además, se propone la tecnología Wi-Fi como la mejor elección para el intercambio sin cables de datos provenientes de los sensores como posiciones, velocidades, direcciones, temperaturas y demás. También se establece que las distancias entre vehículos no deberían ser grandes para así poder asegurar el correcto funcionamiento.

De esta manera, el trabajo presentado en el presente documento sigue esas líneas de desarrollo, intentando realizar lo que se define como ‘una necesidad de mejorar la seguridad en la carretera’. Se ha conseguido, de hecho, llegar más lejos que la mayoría de los proyectos ya realizados y aportar una implementación en un escenario real.



## **CONTRIBUCIONES**

Consideramos, como también se hace en otras propuestas, que los vehículos están equipados con sensores y dispositivos de comunicación que recopilan y comunican la información extraída de dichos sensores.

En nuestro caso concreto, las situaciones de emergencia son tratadas de una manera diferente: se integra una ontología dentro de cada vehículo para dotarlos con capacidad de razonamiento y así evitar el uso de un punto central.

Nuestro sistema beneficia la organización del tráfico en situaciones de emergencia con decisiones inmediatas en las que cada vehículo es un punto de decisión, reacciona considerando los vehículos que le rodean, y colabora con ellos para llegar a un consenso en tiempo real.

Considerando el caso de las autopistas, la estrategia utilizada para diseñar la ontología es limpiar el carril más rápido, esto se hace intentando mover todos los demás vehículos afectados a los carriles más lentos. De esta manera, el vehículo prioritario tendrá paso libre y será capaz de realizar su servicio en el mínimo tiempo posible.

En la tabla siguiente (Tabla 1) se compara el proyecto actual con otros estudios y trabajos realizados.

	Área de actuación	Información necesaria	Proveedores externos de información	Decisión tomada	Tomador de la decisión	Colaboración entre vehículos
<b>Car2Car CC</b>	Amplia y pequeña	Ninguna	Carretera	Rutas alternativas, Advertencias	Dispositivo sugiere, Conductor decide	Difundir mensajes de emergencia
<b>SimTD</b>	Amplia	Destino, Estudio del tráfico	Infraestructuras	Adaptar las rutas	Dispositivo sugiere, Conductor decide	Difundir mensajes de emergencia
<b>PreVent</b>	Amplia	Posición del vehículo, Información meteorológica, Mapas	Señales de tráfico	Advertencias (dejar paso libre, no cruzar, no cambiar de carril, freno automático)	Dispositivo sugiere. Si el conductor no actúa, el dispositivo lo hace por él	No colaboración (sólo sensores)
<b>FleetNet</b>	Amplia	Localización geográfica	Pasarelas fijas	Advertencias (atascos, freno de emergencia, accidente,...)	Dispositivo informa, Conductor decide	Intercambio de posiciones y peligros
<b>DSRC</b>	Amplia	Mapas	Infraestructuras	Advertencias (situaciones de peligro, frenado con antelación, condiciones meteorológicas, accidentes, condiciones de la carretera, obstáculos,...)	Dispositivo informa, Conductor decide	Difundir mensajes de emergencia
<b>NoW</b>	Amplia	Ninguna (no hay organización del tráfico, sólo advertencias)	Infraestructuras	Rutas alternativas, Advertencias	Dispositivo sugiere, Conductor decide	Difundir mensajes de emergencia
<b>Coopers</b>	Pequeña	Mapa de la carretera, Información monitorizada desde el conductor	Infraestructuras	Velocidades máximas (dependiendo de la situación del tráfico)	Infraestructuras sugieren, Conductor decide	No colaboración (comunicación entre infraestructuras y vehículos)
<b>Anemone</b>	Amplia	Mapa de la red	No	Camino más corto (decisión centralizada)	Dispositivo sugiere, Conductor decide	Compartir la posición
<b>Safespot</b>	Amplia	Mapa de la carretera	Infraestructuras	Advertencias (futuros peligros)	Dispositivo informa, Conductor decide	Difundir mensajes de emergencia
<b>Poseidon</b>	Amplia	Mapa de la red	No	Optimización de rutas	Dispositivo sugiere, Conductor decide	Compartir información sobre la situación del tráfico
<b>Nuestra Propuesta</b>	Pequeña (una carretera)	Posición y velocidad	Meteorología e infraestructuras (líneas futuras)	Sugerencias de movimientos exclusivos para cada conductor individualmente	Dispositivo sugiere, Conductor decide	Decisión consensuada (colaboración entre vehículos)

**Tabla 1: Descripción detallada del trabajo realizado y comparación con otros.**

Como puede verse en la tabla anterior, la mayoría de los proyectos que ya han sido realizados se centran más en áreas amplias que en pequeñas y concretas, incluyendo más de una carretera. Esto significa que el trabajo se hace en términos generales, teniendo en cuenta el flujo del tráfico más que centrándose en cada vehículo individualmente y sus movimientos (en lo que se centran las propuestas del presente proyecto).

La mayoría de ellos necesita obtener información sobre las carreteras, como mapas de carretera, y datos sobre el posicionamiento de los vehículos, lo que va relacionado también con la organización del tráfico, no áreas concretas con los individuos que las ocupan.

Además, tienen la necesidad de recopilar información sobre el escenario de algún sitio para así poder trabajar sobre el mismo, y esta información es extraída de infraestructuras situadas a lo largo de la carretera. En pasos futuros, la información proveniente de dichas infraestructuras y también la relacionada con los fenómenos meteorológicos serán incluidas al sistema, de manera que las soluciones propuestas serán incluso más precisas y adaptadas al escenario concreto que se intenta solventar.

En otras propuestas, las decisiones tomadas para solucionar la situación de emergencia son básicamente dos: expandir los mensajes de alarma o proponer rutas alternativas. En nuestro proyecto, además de difundir la señal de alarma, se dan instrucciones más concretas: cada vehículo individual que se encuentra afectado por la situación excepcional obtiene una serie de sugerencias en forma de acciones personalizadas y exclusivas a realizar.

Las soluciones de los demás trabajos se centran en cómo evitar encontrarse con áreas conflictivas, pero el presente proyecto pone especial atención en cómo solventar las situaciones de emergencia una vez los vehículos ya están afectados por la misma.

Un punto en común entre el trabajo que se explica en este documento y la mayoría de los demás ya realizados es el hecho de que las soluciones propuestas son solamente sugerencias, las cuales el conductor tendrá que recibir, considerar si son correctas o no, y realizarlas en el primer caso. El conductor es el que toma la decisión final, la tecnología solamente sugiere acciones.

Finalmente, el nivel máximo de coordinación entre vehículos en las demás propuestas es el simple hecho de compartir sus posiciones y difundir las señales de emergencia. Este es el punto más fuerte del trabajo realizado en este proyecto: la colaboración entre vehículos para llegar a una solución consensuada.

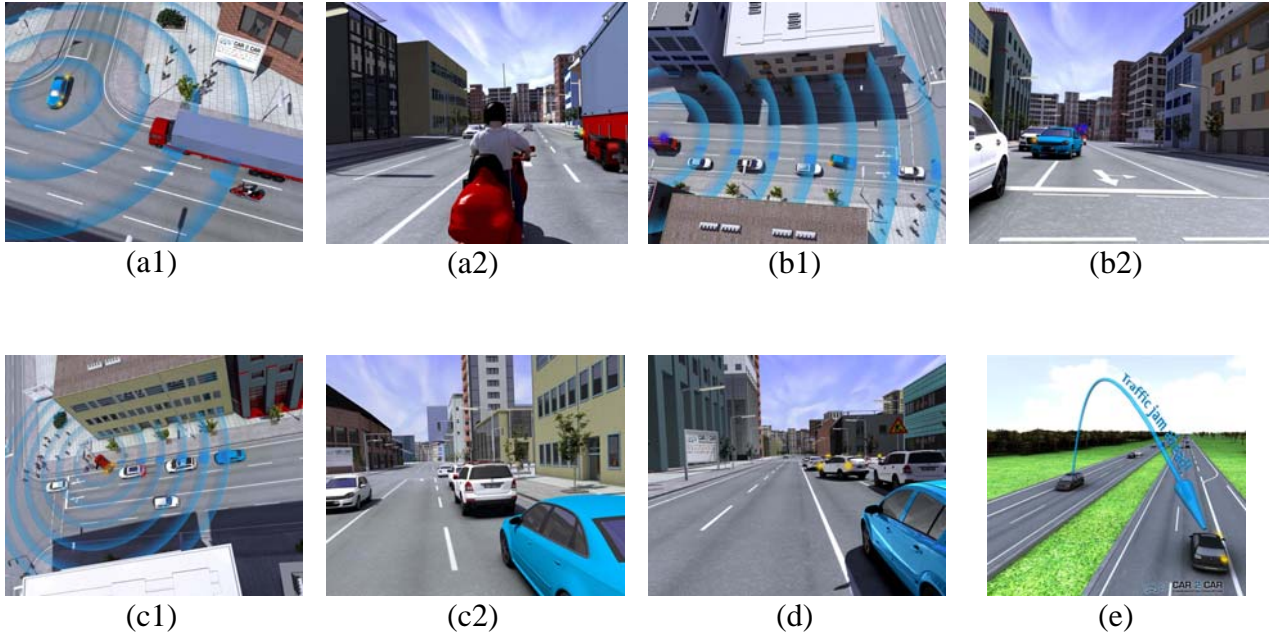
## ***ESTADO DEL ARTE***

A lo largo de los últimos años se han realizado un gran número de proyectos sobre este tema. Todos ellos están relacionados con Vehicular Ad-hoc Networks (comúnmente llamadas ‘redes VANET’), pero la mayoría simplemente exponen los problemas, es decir, explican unos pocos escenarios clave en los cuales serían necesarias esas redes, sin aportar alguna solución que ayude a solucionarlas de una manera u otra.

Algunos de esos proyectos realizados y empresas involucradas en la realización de los mismos son: Car2Car – CC [1], SimTD [2], PReVENT, FleetNet [3], DSRC, SeVeCom [4], NoW [5], Coopers [6], Anemone [7], Poseidon [8] and GST Rescue [17].

En el proyecto [1] se muestran los principales escenarios en los que se producen situaciones de emergencia. Son claramente explicados y retratados, pero ninguna solución es planteada para la solución del mismo. Estos escenarios son los siguientes:

- ✓ Advertencia de motocicleta (obstrucción de visión). (Ilustración 1, a1 y a2)
- ✓ Aproximación de vehículo en emergencia. (Ilustración 1, b1 y b2)
- ✓ Advertencia de luces de emergencia (obstrucción de visión). (Ilustración 1, c1 y c2)
- ✓ Advertencia de obras en la carretera (obstrucción de visión). (Ilustración 1, d)
- ✓ Evasión de atascos. (Ilustración 1, e)



**Ilustración 1: Casos de uso (Car2Car Communication Consortium).**

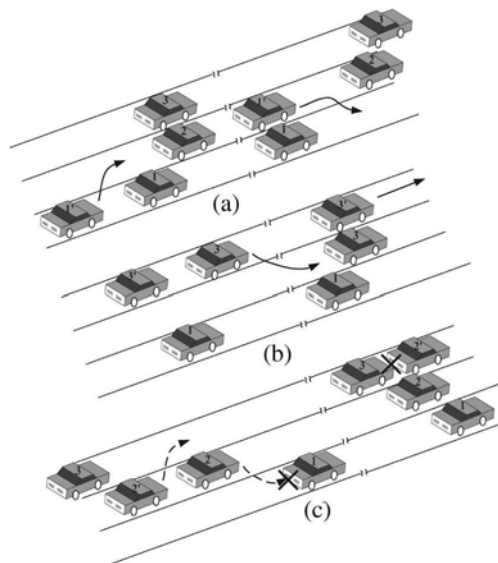
La mayoría de los trabajos realizados están más centrados en el estudio de la situación de las carreteras, obteniendo información sobre la congestión del tráfico, como [9], y ofreciendo al conductor posibles rutas alternativas, como [10] y [11], para así evitar que los vehículos se vean afectados por atascos o tráfico lento a lo largo del camino. En gran número de casos la información es recolectada en un ente central, el cual toma una serie de decisiones que son transmitidas más tarde a los vehículos afectados, de manera que así actúen correctamente para evitar retrasos en el servicio que están ofreciendo.

Otra investigación realizada, ésta con varias soluciones aportadas para situaciones concretas, es la presentada en [12]. Los vehículos que se encuentran en situación de emergencia informan a un punto central sobre la situación. Éste realiza una serie de computaciones y después disemina varias órdenes básicas hacia los otros vehículos que se encuentran presentes en la trayectoria que seguirá el vehículo prioritario.

En este caso particular, la capacidad de razonamiento está alojada en el punto central, y los demás vehículos son notificados sobre la situación y las acciones que tienen que realizar antes de la llegada del vehículo en emergencia a sus posiciones. De esta manera se consigue advertir con suficiente antelación del peligro para que los conductores afectados tengan suficiente tiempo para reaccionar de manera correcta y sin tomar ningún tipo de riesgo.

Semáforos y demás señales de tráfico, así como otros aparatos situados a lo largo de la carretera, bajo el control de dicho punto central, están continuamente recopilando y diseminando información actualizada desde y hacia los vehículos que circulan en ese momento por la misma. Además, dichas señales de tráfico están configuradas por el punto central inteligente para poder así controlar y distribuir el tráfico en esas situaciones de emergencia.

Se han realizado también varias pruebas en escenarios reales, [12] y [13], y los resultados obtenidos han sido bastante satisfactorios. En [13] un escenario concreto ha sido analizado y estudiado: una carretera recta con tres carriles en la misma dirección y sentido, y solamente tres posibles cambios para cada vehículo ('Vehículos adelantando hacia los carriles lentos'; 'Dejar paso a vehículos que circulan con mayor velocidad'; 'No cambio de carril').



**Ilustración 2: Efecto de situaciones dispersas y densas sobre patrones de movilidad.**

- (a) Posibilidad de adelantar, mediante una maniobra simple hacia la izquierda, de un vehículo más rápido.**
- (b) Posibilidad de adelantar, mediante una maniobra simple hacia la derecha de un vehículo más lento.**
- (c) Imposibilidad de adelantar ni de maniobrar.**

Las propuestas actuales consideran que los vehículos están equipados con sensores y aparatos de comunicación que recopilan y dispersan información sobre los sensores instalados en los mismos y en el entorno. Sin embargo, dichos vehículos no integran ninguna capacidad de razonamiento, trabajo que se deja a un punto central.

La organización y distribución del tráfico se podría beneficiar de decisiones inmediatas, en tiempo real, evitando el uso del nodo central. En dicho caso, cada vehículo recogería su propia percepción del escenario a solventar, al cual agregaría la captación de la situación facilitada por los demás vehículos cercanos a él.

El uso de una ontología común a todos los vehículos asegura una rápida y efectiva respuesta a los eventos. Como todos los vehículos comparten la misma ontología, todos ellos llegan a un consenso y actúan de manera conjunta evitando realizar acciones inconsistentes, que puedan perjudicar a los demás o que sean incoherentes entre sí.

Todas las características explicadas y discutidas en este apartado están resumidas en la tabla que se muestra a continuación (Tabla 2):

	<b>REQUISITOS</b>	<b>PROPUESTAS ACTUALES</b>
<b>PROBLEMA ESTUDIADO</b>	Organizar cada vehículo en situaciones de emergencia	Evitar congestión del tráfico y estudiar escenarios peligrosos.
<b>ÁREA DE ACTUACIÓN</b>	Área pequeñas, normalmente no mayor de una calle o carretera	Áreas amplias con varias carreteras y calles
<b>TECNOLOGÍAS DE POSICIONAMIENTO</b>	GPS + Otras tecnologías sin cables	Solamente GPS
<b>SOLUCIONES APORTADAS</b>	Soluciones detalladas para cada vehículo individualmente	Soluciones no aportadas o demasiado simples y generales

**Tabla 2: Comparación general con otros trabajos realizados.**

## **CAPÍTULO 2: HERAMIENTAS**

En este capítulo se explican detalladamente todas y cada una de las herramientas que han sido necesarias para poder realizar el presente proyecto. Se presentan en el mismo tanto las herramientas software como las herramientas hardware, en las que han sido incluidas las primeras.



Para la realización del presente proyecto se ha necesitado principalmente una sola herramienta, que ha sido un ordenador portátil. No era necesario un gran despliegue tecnológico y, además, sus características técnicas no tenían por qué ser muy buenas, con un sistema básico ha sido más que suficiente. Se ha utilizado un simple ordenador portátil, ya que no se ha trabajado ni con una gran carga de gráficos, ni ha habido una necesidad de un nivel de procesamiento alto.

El trabajo realizado con el mismo ha sido básicamente de investigación y de generación de código. La investigación se ha fundamentado en la realización de búsquedas en la web y el desarrollo se ha basado en la creación de una ontología (sistema de toma de decisiones) y de una aplicación que actúa como prototipo para mostrar el funcionamiento de dicha ontología.

El sistema operativo instalado en el ordenador de trabajo ha sido uno bastante común y utilizado por la sociedad, Windows XP, que ha permitido el uso de las herramientas del software sin ningún tipo de problema o incompatibilidad.

Para la creación de la ontología anteriormente citada se ha utilizado una herramienta concreta de desarrollo y manejo de las mismas: Protégé, en su versión 3.4.7. Se han probado también versiones más nuevas a la finalmente elegida (entre ellas la más actualizada, la 4.2) y, aunque todas ellas son muy similares, se han descartado porque las funcionalidades requeridas no se ajustaban a las expectativas puestas en ellas.

Se ha necesitado un tiempo de aprendizaje de esta herramienta, ya que era totalmente desconocida al inicio del proyecto. Para ese proceso de aprendizaje se ha utilizado el manual que la propia empresa creadora de la herramienta pone a disposición de todo usuario en su página web. ([http://owl.cs.manchester.ac.uk/tutorials/protegeowltutorial/resources/ProtegeOWLTutorialP4\\_v1\\_1.pdf](http://owl.cs.manchester.ac.uk/tutorials/protegeowltutorial/resources/ProtegeOWLTutorialP4_v1_1.pdf))

Dicho manual se titula 'A Practical Guide to Building OWL Ontologies Using Protégé 4 and CO-ODE Tools' y, como puede verse, está dirigido a la utilización de una de las últimas versiones de la aplicación. Sin embargo, como se ha comentado anteriormente, todas ellas son prácticamente iguales y el manual de una se puede aplicar sin ningún problema las demás.

Se ha utilizado también una herramienta para el desarrollo de aplicaciones, llamada Netbeans, en su versión IDE 7.0. Con la misma, se ha creado una aplicación prototipo que sirve para mostrar el funcionamiento de la ontología de una manera sencilla e intuitiva. Esta aplicación está escrita en lenguaje de programación Java.



**Ilustración 3: Protégé.**



**Ilustración 4: Java Platform.**

Al principio del proyecto, cuando todavía no se había empezado ni siquiera a investigar sobre el estado del arte, se tomó un pequeño período de toma de contacto con el proyecto en general. Es decir, se realizó una labor de comprensión del proyecto global del cual forma parte el que en este documento se describe.

Se estuvieron haciendo una serie de pruebas sobre canales de transmisión, anchos de banda y transmisiones punto-a-punto, teniendo en cuenta los niveles de ocupación de dichos canales y el ruido encontrado, el cual dificulta las transmisiones entre los vehículos. Para todo ello se utilizó una herramienta



concreta que facilitó la labor de recogida y posterior estudio de los datos obtenidos en dichas pruebas:  
Ubiquiti Airview Spectrum Analyzer Software. (<http://www.ubnt.com/airview>)

## CAPÍTULO 3: REQUISITOS

En este capítulo se especifica con todo detalle una amplia lista de requisitos del proyecto. Se presentan los requisitos funcionales, los cuales tratan sobre qué debe hacer el producto obtenido del desarrollo del proyecto y cómo. También están los requisitos técnicos y los requisitos tecnológicos, que proporcionan unas pautas más detalladas sobre la realización del proyecto.

Finalmente se muestran una serie de diagramas de casos de uso que detallan visualmente las funcionalidades requeridas.

Para la realización del presente proyecto no se han impuesto muchos requisitos, se ha dado bastante libertad y apertura a diferentes posibilidades. Todo esto ha sido así porque para el proceso de investigación y de desarrollo posterior no se han establecido muchas pautas sobre qué tecnologías, herramientas o técnicas utilizar, por lo que si durante el desarrollo se encontraba algo novedoso y/o mejor de lo que se estaba utilizando hasta el momento, se podía integrar en el proyecto.

Solamente los aspectos básicos están condicionados por los requisitos impuestos al inicio del desarrollo del presente proyecto, como las ideas generales de lo que se quiere realizar. Todo lo demás, relacionado con las herramientas a emplear o la manera de hacerlo, entre otros, ha sido diseñado para dar total libertad al desarrollador a la hora de decidir qué utilizar y cómo hacer las cosas.

Estos requisitos establecidos al inicio del proyecto se desglosan y explican detalladamente a continuación, divididos en tres grupos bien diferenciados:

### ***REQUISITOS FUNCIONALES:***

Hay una serie de requisitos que marca la línea principal del proyecto y establecen la base del mismo. Especifican detalles sobre qué desarrollar, y se desglosan a continuación:

- Se pretende dar una solución a situaciones de emergencia que se pueden suceder en la carretera. Una situación concreta de entre todas ellas es la que acapara el centro de atención del presente proyecto: vehículo prioritario en situación de emergencia necesita paso libre para poder realizar su trabajo de la manera más rápida posible.
- Las soluciones que se encuentren tras el estudio de la situación deben ser específicas para cada vehículo afectado, de manera que cada uno de ellos reciba una información concreta, tratándose cada vehículo de manera individual. Así, cada uno recibe una solución exclusiva, solamente aplicable por él mismo.
- Tanto el área de recopilación de información sobre los vehículos afectados por la situación, como de aportación de soluciones debe ser concreta y específica. No se debe de tratar con varias carreteras o calles de forma simultánea, no debe de haber grandes distancias entre los vehículos afectados.  
Todo esto es así porque, como se van a estudiar las situaciones individuales de cada vehículo, sería inviable realizar esto para un número muy alto, ya que supondría un coste computacional implausible.
- La solución encontrada se debe transmitir al conductor de una manera concreta, sencilla y, sobre todo, fácil de entender y de llevar a cabo. Los tiempos de reacción deben minimizarse para que la situación se solventa de la manera más rápida posible. Por lo tanto, el conductor debe ser capaz de entender el mensaje al momento de recibirlo para poder reaccionar correctamente sin ningún tipo de duda o desconfianza.
- Se pretende que la solución aportada modifique en la menor medida de lo posible el flujo normal del tráfico. A menor número de cambios requeridos para solventar la situación de emergencia, menor impacto negativo tendrá sobre el flujo normal del mismo.

- Un requisito básico que se exige a las soluciones aportadas, y que si no se cumpliera el proyecto no tendría ningún sentido, es que se mantenga en todo momento la seguridad en la carretera de todos sus ocupantes.

La solución encontrada no debe suponer la creación de nuevas situaciones de peligro ni incómodas para el conductor, ya que, si fuera así, en vez de solucionar una situación de emergencia lo que se haría sería crear nuevos peligros.

- Otro requisito que se establece para la realización del proyecto, y quizás el más importante de todos ellos, es que se establezcan comunicaciones entre los diferentes vehículos afectados por el escenario. De esta manera se puede buscar una solución común a todos, que altere su tránsito en la menor medida de lo posible.

Este proyecto está basado en la comunicación, es decir, sin ella no se podría llegar a tomar las decisiones que solventen la situación de peligro. Estas comunicaciones son primordiales a la hora de compartir información individual de cada vehículo, de manera que se pueda tener una visión completa del escenario a evaluar.

- Los datos a compartir entre vehículos en esas comunicaciones son un tema bastante abierto, es decir, solamente hay un único requisito sobre ellos: que se retransmita la señal de alarma.

El desarrollador es el encargado de definir la información a compartir sobre cada vehículo presente en el escenario, la cual se utilizará para la posterior toma de decisiones.

## ***REQUISITOS TÉCNICOS***

Además de los requisitos funcionales, también se establecen una serie de condiciones sobre los detalles técnicos del proyecto:

- Se exige que el sistema en el que se base toda la comunicación, así como la toma de decisiones, sea distribuido. Se quiere evitar el uso de un nodo central que actúe como punto de recopilación y salida de información hacia los demás elementos de la red formada por los vehículos.
- Además, la toma de decisiones debe ser lo más rápida posible, para así asegurar que se pueden solventar las situaciones de emergencia en el momento en el que se producen. Si encontrar dicha solución supusiera un tiempo considerable, no merecería la pena utilizarlo ya que, para cuando se vaya a poner en marcha, el escenario habrá sufrido modificaciones y no tendrá ningún sentido aplicar las soluciones propuestas.
- El sistema de toma de decisiones debe ser también dinámico, es decir, se debe adaptar a los cambios en el flujo del tráfico y además debe ser capaz de detectar los movimientos de cada vehículo individualmente para así tener en cuenta si las acciones propuestas se van realizando o no. De esta manera, al llevar un control de cómo se va solventando la situación de emergencia, se pueden adaptar mejor las soluciones sugeridas a los conductores.

Todo movimiento será detectado, por lo que el tráfico va a estar monitorizado de manera continua para ajustar las soluciones dadas a las reacciones de los conductores.

## **REQUISITOS TECNOLÓGICOS**

A pesar de que no se han establecido apenas requisitos relacionados con la tecnología a utilizar, sí que se ha producido un pequeño encaminamiento hacia ciertos aspectos concretos:

- Se ha propuesto la utilización de ontologías como método más apropiado para la creación y el manejo de los sistemas de toma de decisiones. Esto ha sido así principalmente porque se trata de una tecnología novedosa que se está empezando a utilizar cada vez con más frecuencia, y que además aporta una nueva dimensión a los sistemas expertos que se han utilizado hasta el momento.

Para ello se requiere el conocimiento de una aplicación concreta para el manejo de dichas ontologías: Protégé.

- Finalmente, después de todo el estudio y creación relacionados con las ontologías y la toma de decisiones, se requiere la construcción de una aplicación prototipo que muestre de una manera clara y sencilla el funcionamiento de la ontología desarrollada.

Si no se crea este prototipo puede ser realmente complicado realizar una simple comprobación del correcto funcionamiento de la ontología. También, para una persona que no esté habituada al trabajo con el tema tratado, puede ser realmente complicado comprender la ontología si se intenta mostrar directamente con las herramientas de desarrollo de la misma.

## **CASOS DE USO**

Los casos de uso que se muestran en el Apéndice B están basados en los requisitos funcionales detallados anteriormente, y describen las actividades que se deben llevar a cabo para realizar los procesos necesarios para el funcionamiento del sistema desarrollado, el simulador.

## CAPÍTULO 4: ANÁLISIS

En este capítulo se realiza un análisis profundo de la situación en la que se engloba el proyecto, así como del hecho de utilizar una ontología. Además, se analiza también la selección del lenguaje de programación y del simulador que se construirá con el mismo, y que será el que muestre el funcionamiento real de todo el sistema desarrollado en el presente proyecto.

Finalmente, se muestran una serie de diagramas de secuencia, las cuales aportan una visión de cómo van a ser las comunicaciones dentro del sistema, y un diagrama de clases.

## ANÁLISIS DE LA SITUACIÓN

Después de todo el estudio de lo que ya está desarrollado y que está siendo desarrollado sobre el tema de la seguridad en la carretera, se han extraído una serie de conclusiones y se han tomado las decisiones pertinentes, que son las que han marcado cómo se va a desarrollar el proyecto y porqué se va a realizar de esa manera.

Definitivamente se ha decidido por centrarnos en un aspecto concreto de todos los que están englobados por el término ‘seguridad’, el escenario en el que un vehículo en emergencia necesita paso libre lo más rápido posible. La elección del mismo se ha realizado porque se trata de una situación que todos los ciudadanos nos podemos encontrar en nuestra vida diaria, y de hecho nos encontramos con bastante frecuencia.

Las experiencias personales señalan que hay mucha incertidumbre en estas situaciones, y que debería existir algún mecanismo que ayude al conductor a tomar las decisiones correctas para facilitar el paso del vehículo prioritario.

Se ha establecido que el sistema sea distribuido, sin la presencia de un ente central que controle todo y por el cual tenga que pasar toda la información para la toma de decisiones. Esto ha sido así porque si no se especularía con nuevas dudas a las que hacer frente: ¿Qué es ese punto central (un vehículo, un edificio, las señales de tráfico)? ¿Dónde se sitúa? ¿Qué carga puede soportar? ¿Cuánto dinero cuesta?...

Además de todo esto, se ha considerado que un sistema descentralizado es el más adecuado para el proyecto actual por varias razones, como:

- Facilidad de implantación.
- Mejores prestaciones.
- Materiales más económicos.
- Simplicidad,...

El hecho de centrarse en áreas de tamaño controlado está relacionado con el control que se hace sobre cada vehículo individualmente. Si se tratasen los vehículos por grupos o zonas no se podría estudiar el caso de cada uno de ellos por separado, por lo que las sugerencias que se les aportarían no serían lo suficientemente precisas y exactas para la conseguir solventar la situación de emergencia de manera eficiente.

Para buscar dichas soluciones se ha seguido una estrategia concreta: intentar liberar el carril más rápido para el paso del vehículo prioritario, mediante la incorporación de los demás vehículos involucrados a los carriles más lentos. Esto se ha establecido así porque a un vehículo que se encuentra en situación de emergencia, y que probablemente circule a una velocidad mayor de lo normal, no se le puede exigir que zigzaguee entre los vehículos para alcanzar su destino. Además, es posible que dentro del mismo se esté desempeñando algún trabajo que requiera un mínimo de precisión y calma, como por ejemplo si se encuentra un paciente dentro de una ambulancia, que está siendo trasladado y a su vez se le está tratando.

Una regla establecida para la toma de decisiones es que, en caso de poder elegir entre decelerar o acelerar, siempre se elige la segunda opción, principalmente sobre los vehículos que se encuentran en el carril que tiene que liberarse. De esta manera no se entorpece el paso del vehículo

en situación de emergencia y éste va avanzando de manera progresiva, evitando que tenga que reducir su velocidad en algún momento.

El hecho de buscar una solución consensuada y no simplemente diseminar el mensaje de emergencia está basado en la idea de que, con una colaboración y decisiones acordadas, los resultados son mucho más exitosos que si se pide a cada conductor que actúe de la manera que le parezca más correcta. De este modo, todos actúan de manera conjunta, sin incoherencias y de una manera efectiva, evitando la incertidumbre lógica sobre cómo reaccionar cuando un conductor recibe un mensaje de emergencia.

Finalmente, la solución a la cual se ha llegado y que se comunica al conductor no es una orden, se trata de una sugerencia. Esto es así porque, a pesar de que se ha decidido tras un razonamiento inteligente que esa es la mejor opción, una persona se supone con suficiente sentido común como para ser capaz de decidir si es correcto realizarla o no, basándose en lo que percibe a través de sus sentidos. Esto no supone que lo sugerido carece de valor, sino que ambas partes son necesarias y se complementan, ya que el conductor solamente conoce una pequeña parte del área afectada y los sistemas de decisión no tienen adquirido el sentido común del ser humano. La combinación de los dos elementos es la fórmula que solventa el problema.

Por todo ello, la acción a realizar por el conductor no se automatiza y la ejecuta el vehículo de manera independiente, sino que es necesario el razonamiento de una persona. Además, quizás la sociedad no esté preparada para dejar en manos de la tecnología la conducción de los vehículos, y con ello su integridad física.

## ONTOLOGÍA

Se ha decidido que la creación y el manejo del sistema de toma de decisiones estarán basados en ontologías. En los últimos años se está comenzando a dar importancia a este tipo de herramientas y cada vez se están implantando más en nuevos sistemas y aplicaciones que se desarrollan en la actualidad. Todavía necesita cierto tiempo para que se vaya asentando y madurando de manera definitiva como una herramienta con un uso extendido, pero las expectativas son muy altas.

Además de todo ello, la elección de la utilización de ontologías se ha realizado porque proporciona total independencia entre razonador y programación. Es decir, los razonamientos que utiliza una aplicación se realizan en un lugar externo, de la ontología, lo que implica que en el momento que sea necesario realizar modificaciones tanto en un sitio como en el otro, no supondrá un arduo trabajo. Los cambios se realizarán de una manera muy simple y rápida, ya que no hay dependencia entre ambas partes, evitando así propagación de errores, pérdidas de tiempo en encontrar lo que se busca, etc.

La ontología ha sido diseñada de una manera bastante general, de manera que se trabaja con parámetros, en vez de valores. Así se puede manejar y adaptar de una manera más simple a cambios que se puedan producir por el avance del proyecto, añadiendo nuevas funcionalidades y puliendo las que ya están incluidas.

Una vez decidido que las ontologías serían la base de la inteligencia del proyecto, se eligió Protégé como aplicación para el manejo y la creación de las mismas. Esta elección fue basada en que no había ningún tipo de conocimiento sobre este tipo de herramientas, el tutor del proyecto



propuso la misma, la mayoría de los trabajos realizados parecen estar desarrollados con Protégé, y también el hecho de que se disponía de manuales y tutoriales, además de la ayuda de varios compañeros, que trabajaban en otras partes del proyecto global y que ya tenían ciertos conocimientos sobre la herramienta y su funcionamiento.

Se probaron varias de las últimas versiones de esta aplicación, como por ejemplo la más reciente (4.2), se hicieron pruebas con todas ellas y finalmente se optó por utilizar la versión 3.4.7. Esto fue basado en que había varios funcionamientos indeseados de las versiones más recientes a la misma relacionados con las reglas que se utilizan para extraer las soluciones, en las cuales se basa todo el razonamiento de la ontología. En dichas versiones los resultados no se mostraban, o lo hacían de una manera poco correcta e incomprensible. Por lo tanto, se eligió la 3.4.7 porque todas las funcionalidades deseadas se realizaban de manera clara y correcta, con un formato que daba bastantes facilidades a la hora de la creación de dichas reglas y supervisión de los resultados.

## ***SIMULADOR - LENGUAJE DE PROGRAMACIÓN***

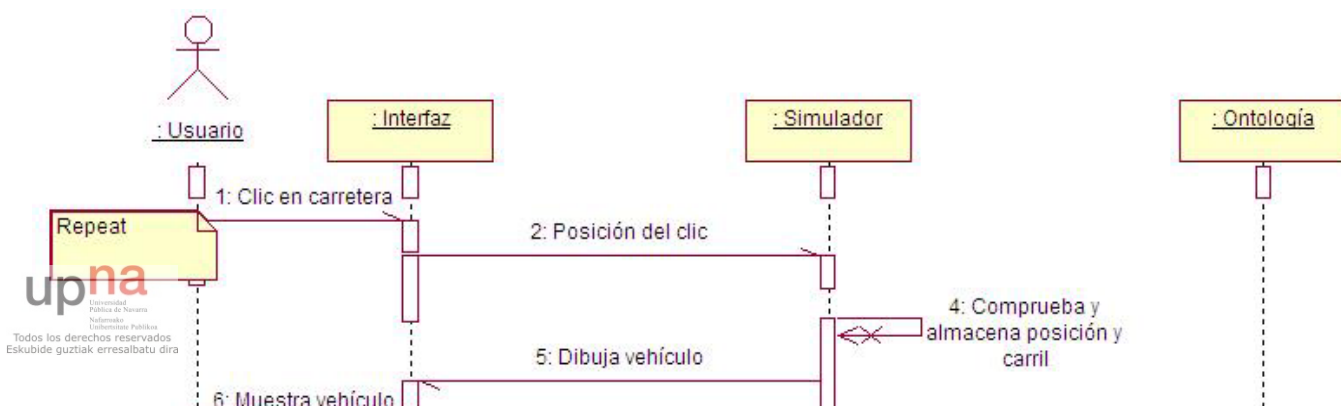
Tal y como se ha comentado en páginas anteriores, se ha desarrollado una aplicación prototipo, que utiliza la ontología creada y simula el proceso de razonamiento para extraer las decisiones a tomar por los conductores. Estas soluciones son directamente aplicadas sobre el escenario presentado en la aplicación, de manera que se puede percibir fácilmente cómo se va subsanando de manera progresiva la situación de emergencia.

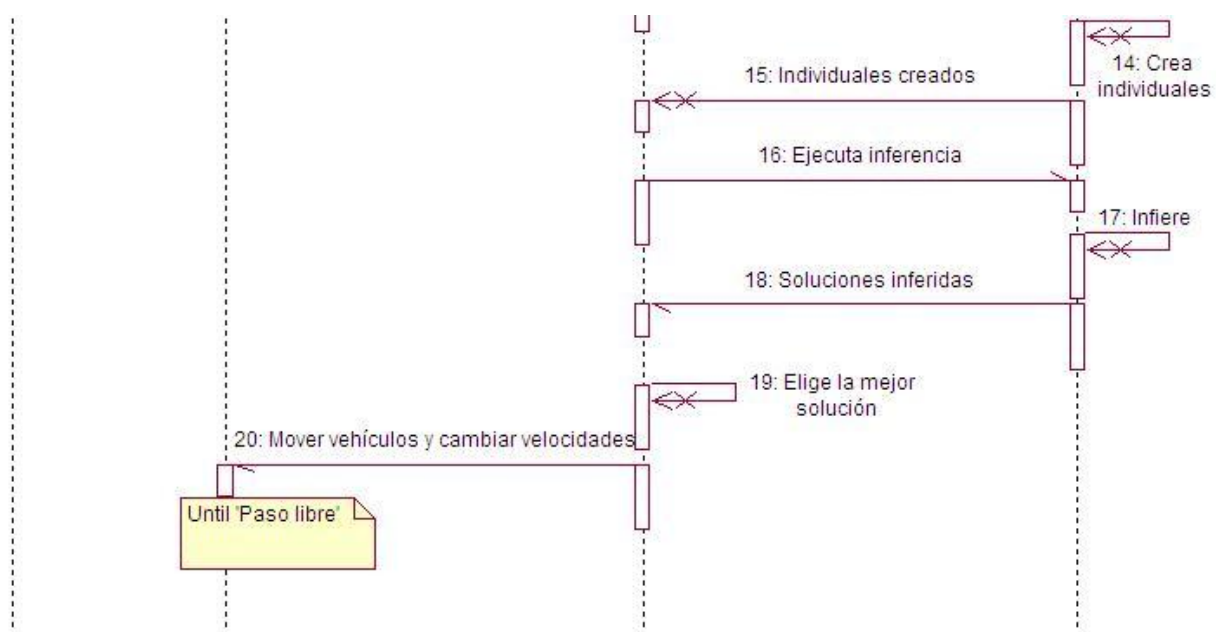
Dicho simulador ha sido construido en lenguaje de programación Java. Fue la primera elección realizada a la hora de desarrollar software, y fue seleccionado de manera aleatoria, ya que para realizar una simple aplicación como la desarrollada no es necesario un gran despliegue ni características muy específicas, por lo que Java era más que apto para ello.

Poco más adelante en el proyecto, cuando se comenzó con la integración de la ontología en dicha aplicación, se corroboró que la elección de ese lenguaje de programación era la idónea. Esto es debido a que todo lo relacionado con las ontologías, y Protégé en concreto, está encaminado hacia Java. Todos los tutoriales y manuales consultados, así como las APIs utilizadas para dicha integración están basados o dirigidos al desarrollo en Java, por lo que hay cierta obligación a desarrollar el software en este lenguaje.

La utilización de la aplicación desarrollada es bastante simple e intuitiva. Está todo ya pre-programado para que el usuario no tenga ninguna duda sobre su utilización desde la primera toma de contacto. Además, los resultados son claramente visibles, tanto las decisiones tomadas por los vehículos como su realización visual, traducida en los movimientos de los vehículos sobre la carretera. Todo el proceso es perfectamente comprensible por cualquier usuario.

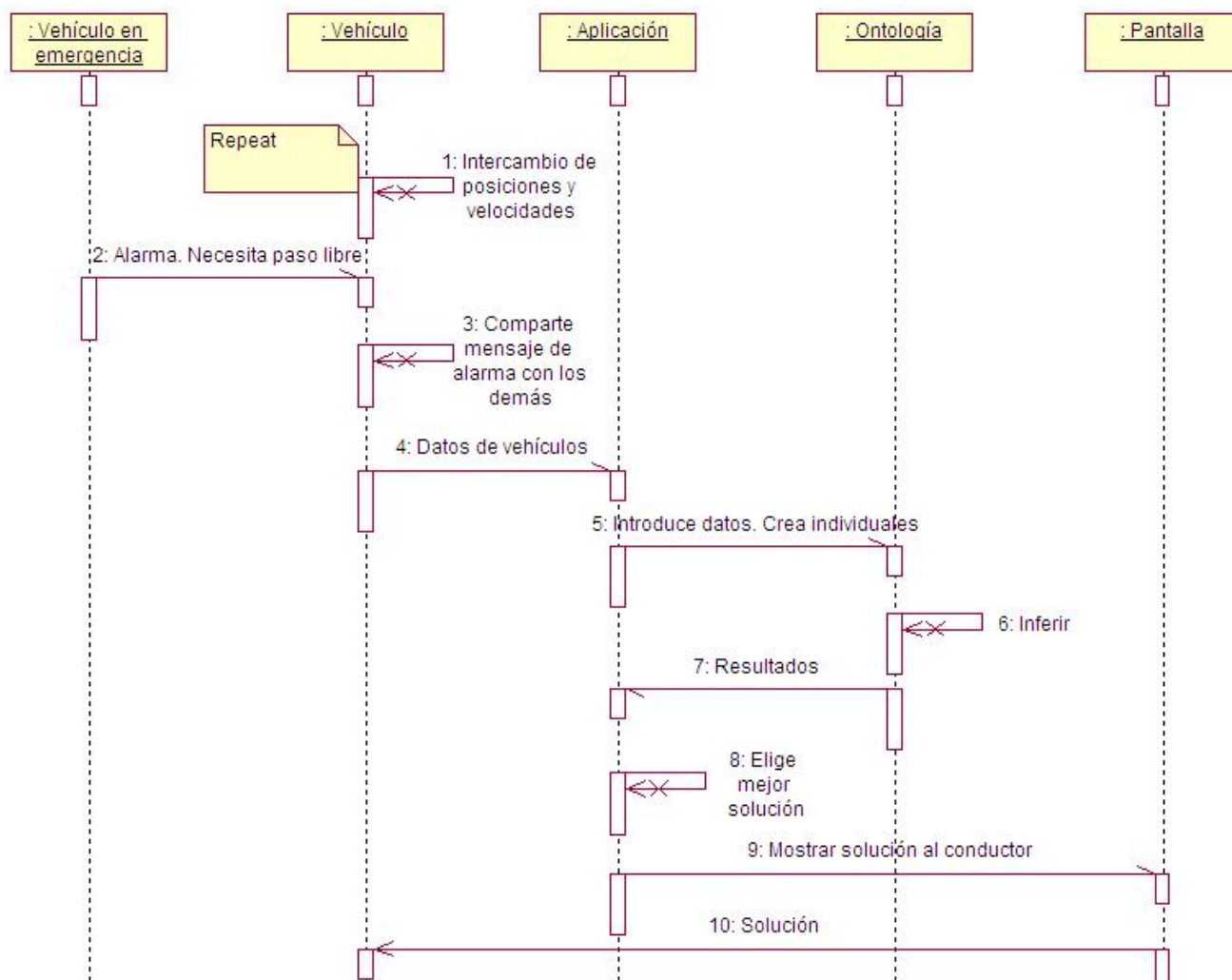
El funcionamiento de dicha aplicación está descrito en el siguiente diagrama de secuencia (Ilustración 5), en el cual se detallan todas las comunicaciones que se establecerían entre los diferentes actores y elementos del sistema:





**Ilustración 5: Diagrama de secuencia - Simulador.**

Y el funcionamiento que se obtendría sobre un escenario real se muestra en el siguiente diagrama de secuencia (Ilustración 6):



**Ilustración 6: Diagrama de secuencia - Escenario real.**

Los diagramas de colaboración relacionados con los diagramas de secuencia anteriores se muestran en el Apéndice C.

## CAPÍTULO 5: DISEÑO

En este capítulo se presenta la arquitectura que tiene el sistema desarrollando, relacionado tanto con el simulador construido como con la implantación en un entorno real. Además, se presentan también las clases que forman parte de dicho simulador.

## ARQUITECTURA DEL SISTEMA

El sistema desarrollado en el presente proyecto tiene una arquitectura bastante simple, la cual muestra el diseño y el funcionamiento del mismo.

A continuación se presentan dos arquitecturas distintas sobre el mismo sistema. Esto es así porque la primera representa el escenario real, lo que sería la implantación en sistemas hardware, con la instalación de los dispositivos necesarios, y el funcionamiento que se prevé del sistema. Este escenario representa el objetivo que se persigue con la realización del proyecto.

Por otro lado, se presenta también la arquitectura del sistema basada en la utilización del simulador, ya que todavía no se ha podido realizar la instalación completa en un entorno real. Es la herramienta que ha servido para realizar todo el diseño y las pruebas hasta llegar a este punto, en el que el sistema está completamente desarrollado y probado. Solamente quedaría el paso de la instalación para que se diera por completamente funcional.

### a. APLICADA AL ESCENARIO REAL

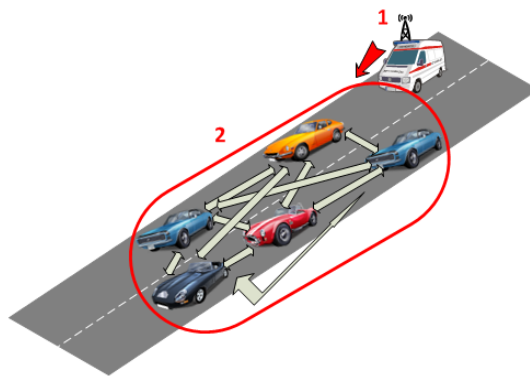
La arquitectura del sistema está basada en el router que lleva incorporado cada vehículo, el cual sirve, tanto para la comunicación con los demás y con una tableta, como para la toma de decisiones, ya que lleva incluida la ontología y el razonador que hacen ese trabajo.

Además de este router, cada vehículo lleva una antena instalada en la parte exterior del techo del mismo, la cual permite transmitir y recibir información.

Por último, cada uno de ellos también incluye una tableta que muestra por pantalla al conductor de manera visual las sugerencias de acciones a realizar.

El proceso que se sigue para la distribución del tráfico y dejar paso libre al vehículo en situación de emergencia (ambulancia, en este caso) es descrito a continuación:

- i. El vehículo en emergencia envía un mensaje de alarma hacia todos los demás vehículos que se encuentran en la carretera, pidiendo paso libre.

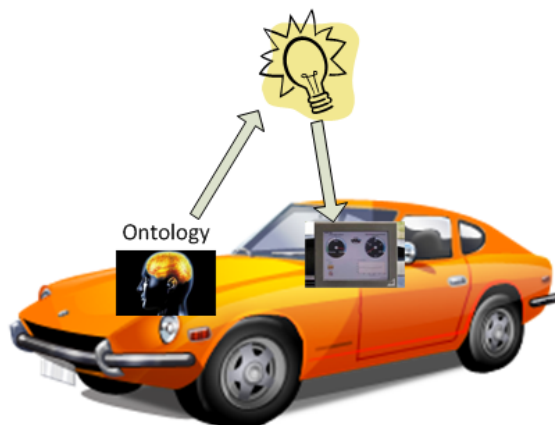


**Ilustración 7: Primeros pasos en la comunicación.**

- ii. Como todos los vehículos se envían información de manera continua, incluso sin la necesidad de que se produzca ningún evento en la carretera, ya se conocen las posiciones y velocidades de cada uno de ellos.

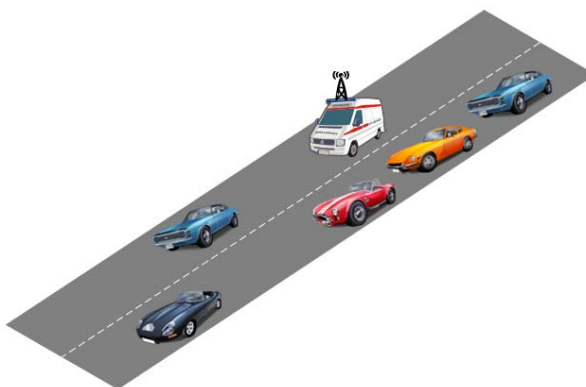
Por lo tanto, cada uno de ellos realiza su propio razonamiento interno para extraer las acciones a realizar, que consiste en:

- Ejecutar la ontología incorporada dentro del router, con todos los datos de los vehículos.
- Extraer las soluciones aportadas por la misma.
- Mostrarlas como sugerencias a través de la pantalla instalada en el interior del vehículo.



**Ilustración 8: Proceso en el interior del vehículo.**

- iii. En este punto, si cada vehículo ha realizado los movimientos que se le han sugerido a través de la pantalla, se produce un progreso en el escenario. En el ejemplo mostrado un vehículo se ha movido al carril lento, dejando así paso al vehículo prioritario. También se ha facilitado el siguiente paso, ya que se ha ido creando un hueco para que se incorpore al mismo el otro vehículo que entorpece a la ambulancia.



**Ilustración 9: Progreso conseguido.**

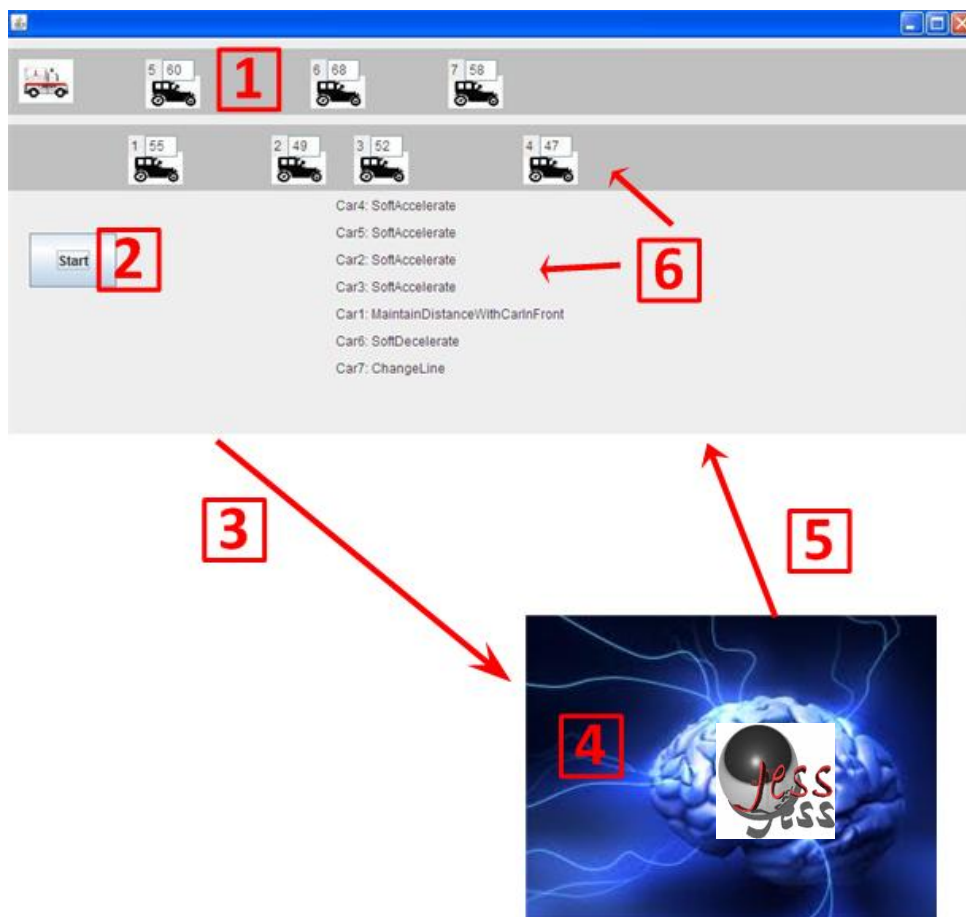
- iv. Los dos últimos pasos se ejecutan de manera repetitiva, hasta que el carril por el que circula la ambulancia se libera totalmente, permitiendo su paso.

## **b. APLICADA AL SIMULADOR**

Esta arquitectura es bastante sencilla, ya que se basa en una aplicación desarrollada en lenguaje Java, una ontología con toda la inteligencia incorporada, y un razonador que es capaz de utilizar esa ontología y extraer unos resultados acordes a la información aportada por parte de la aplicación.

El proceso que se realiza para simular una situación de emergencia en la que se debe dejar paso libre a una ambulancia es el siguiente:

- 1) Se posicionan los vehículos sobre la carretera dibujada en el simulador. Se debe asignar también una velocidad de circulación a cada uno de ellos.
- 2) Cuando se han dispuesto todos los vehículos que se desean con sus correspondientes velocidades, se pulsa el botón 'Start', para que comience el proceso de simulación.
- 3) La aplicación inserta los datos del escenario en la ontología.
- 4) El razonador *Jess* realiza el proceso de inferencia y obtiene los resultados, a partir de los datos aportados en el paso anterior.
- 5) Estos resultados son comunicados al simulador.
- 6) El simulador traduce los resultados de la inferencia en movimientos de los vehículos y muestra una lista con lo realizado por cada uno de ellos.

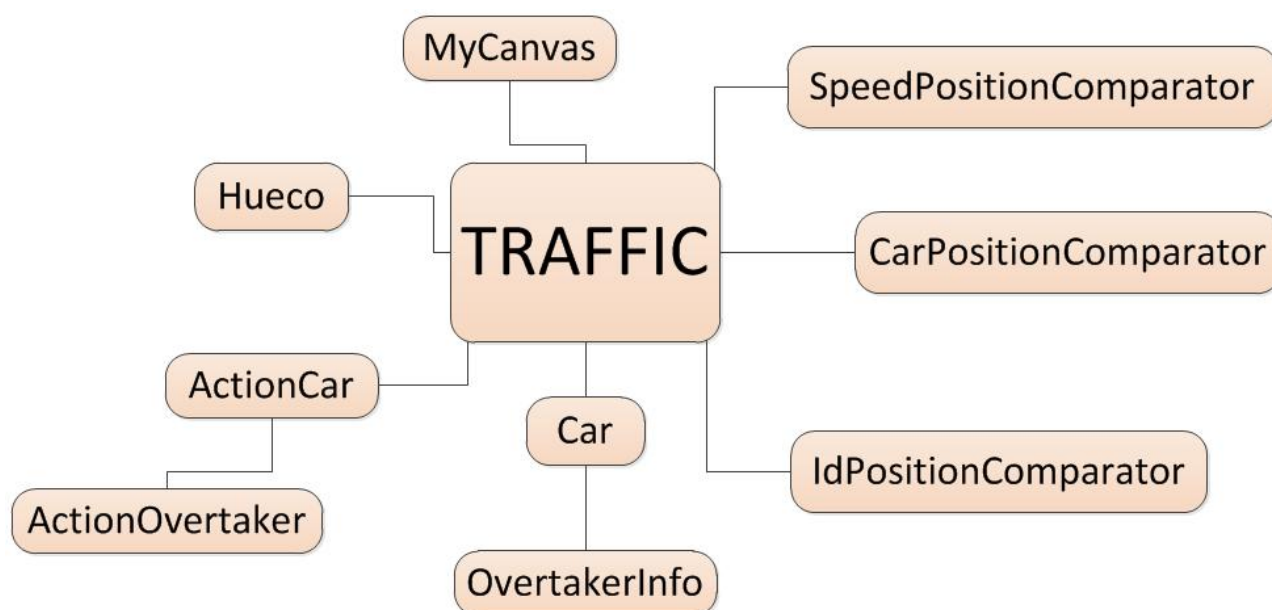


**Ilustración 10: Arquitectura aplicada al simulador.**

## CLASES

La aplicación que se ha construido, y que sirve como simulador del sistema, está compuesta por diferentes clases. Cada una de ellas es designada para realizar una tarea específica, y todas ellas trabajan en conjunto para poder llevar a cabo la simulación.

El diagrama de estas clases puede verse en la Ilustración 11:



**Ilustración 11: Clases que componen el simulador.**

Como se puede apreciar en la figura anterior, todo pasa por la clase central, la cual se encarga de realizar prácticamente todo. Ésta se encarga de realizar la mayoría de los procesos necesarios para el funcionamiento del simulador.

Para la carga del interfaz de usuario, la clase *Traffic* hace uso de la clase *MyCanvas* para dibujar la carretera con sus dos carriles, la ambulancia, y también para posicionar el botón ‘Start’.

Las clases *CarPositionComparator*, *IdPositionComparator* y *SpeedPositionComparator* son utilizadas por parte de la central, en el momento de hacer click en ‘Start’ para poder ordenar las entradas de los vehículos, y sus etiquetas dibujadas en el simulador, dentro de un array que será recorrido varias veces en procesos posteriores.

Estas entradas que se almacenarán en dicho array serán objetos de la clase *Car*, y contendrán toda la información relacionada con cada vehículo y su entorno. Esta información se basará en datos sobre posiciones y velocidades, tanto del propio vehículo como de los que le rodean.

Los datos que cada vehículo tiene en relación con los vehículos que se tienen que incorporar al carril más lento se almacenan en una clase separada de las demás, en *OvertakerInfo*.

Cuando ya se tienen todos los datos necesarios estudiados y organizados, la clase *Traffic* hace uso de la clase *Hueco* para establecer las características de los espacios libres que hay en la carretera y sus asignaciones a los diferentes vehículos que se los van a ocupar.



Finalmente, cuando ya se han extraído todas las acciones a realizar inferidas por la ontología y el razonador Jess, esta clase central lo que hace es estudiarlas todas ellas y asignarlas a cada vehículo (mediante el uso de las clases *ActionCar* y *ActionOvertaker*).

## **CAPÍTULO 6: METODOLOGÍA**

En este capítulo se presenta un desglose temporal sobre cómo va a realizarse cada una de las fases de desarrollo del proyecto, desde la investigación del estado del arte, hasta la adquisición de conocimientos, todas las fases de desarrollo y la presentación del producto final obtenido.

El proyecto desarrollado ha seguido una planificación temporal, basado en un modelo iterativo incremental en el cual se ha ido desarrollando todo el software de manera incremental, partiendo del núcleo de lo que se necesitaba y desde ahí ampliando poco a poco las funcionalidades y la exactitud del mismo. De esta manera, mediante etapas, se ha ido completando el software implementado, realizando cambios tanto en el código como en el diseño.

Así, no ha sido necesario conocer todos los requisitos detallados desde el primer momento, sino que se han ido teniendo en cuenta conforme se ha ido desarrollando el mismo, adaptándolo a las nuevas exigencias, basadas también en lo aprendido durante sus etapas de desarrollo y pruebas. Al no tener tampoco unos requisitos muy estrictos, todos los incrementos, modificaciones y mejoras que se han ido realizando no han supuesto un gran impacto en lo planificado ni anteriormente desarrollado.

Antes de la etapa de inicialización del producto software, ha sido necesario un estudio profundo y exhaustivo del estado del arte, es decir, de todo lo que se ha estudiado, propuesto y realizado con respecto al tema del presente proyecto: la seguridad en la carretera, y concretamente en situaciones de emergencia.

En este paso se han estudiado gran cantidad de proyectos que se han realizado y que se están realizando en la actualidad, relacionados con el tema. Muchos de ellos no estaban directamente relacionados con el caso particular de este proyecto, el dejar paso libre a un vehículo prioritario, pero sí que han aportado información bastante interesante y de gran utilidad para el desarrollo del mismo.

Se ha estudiado qué situaciones trataban cada uno de los proyectos y qué soluciones aportaban. Se han conocido técnicas y tecnologías utilizadas para la extracción de información, la comunicación entre elementos y para la toma de decisiones (si se daba el caso).

También se han conocido las expectativas puestas en este tipo de trabajos que intentan dar solución a las situaciones en carretera y la visión de futuro que se supone de los trabajos que han sido y están siendo realizados.

Más tarde, se ha hecho una crítica sobre lo encontrado, se ha comparado con lo que se pretendía hacer, estudiando los puntos a favor y en contra de todo lo visto y de lo planificado en nuestro proyecto. Se ha pensado sobre la posibilidad de añadir, eliminar o modificar aspectos de nuestro proyecto, basándonos en lo encontrado sobre los otros proyectos, de manera que se pueda desarrollar algo más completo y preciso.

Se ha reflexionado sobre las tecnologías a utilizar (hardware) para poder obtener todos los datos que se suponen necesarios para poder trabajar sobre ellos y desarrollar todo el proyecto en base a ellos. Se han estudiado también las precisiones de los mismos, de manera que sean suficientes para las medidas necesarias en el proyecto, y sus alternativas. Finalmente se ha apartado un poco este tema ya que no incumbe de manera directa el proyecto que explica el presente documento, sino más bien al global del que forma parte.

En este punto, cuando ya se han conocido las características y detalles de todos los proyectos y estudios realizados sobre el tema a trabajar, ha sido el momento idóneo para establecer de manera definitiva las directrices que van a marcar el proyecto que se presenta durante todas sus etapas de realización, suponiendo la base sobre la que se va a desarrollar todo el trabajo.

Entonces, una vez sabido qué se ha a hacer y cómo se va a desarrollar, se han elegido las herramientas y técnicas necesarias para el desarrollo del proyecto que se presenta. También se ha establecido la planificación de las siguientes etapas del proyecto, con todo lo que hay que desarrollar y en qué orden.

Ha sido entonces cuando se ha comenzado a trabajar con las ontologías. No se tenía ningún conocimiento sobre el tema, por lo que se ha necesitado, antes de nada, informarse sobre qué era una ontología.

Una vez ya se han tenido claros los conceptos, se ha comenzado a trabajar con ellos. Se ha descargado (desde la página oficial, <http://protege.stanford.edu/>) e instalado la aplicación para la creación, el desarrollo y el mantenimiento de las ontologías, Protégé, y se ha utilizado un manual para aprender su utilización. Esto ha supuesto un tiempo que se ha empleado a la formación, durante el cual no se ha desarrollado nada directamente relacionado con el proyecto, sino pequeñas pruebas que han servido para adquirir la suficiente práctica y el conocimiento necesarios para poder ser capaz de crear una ontología desde cero y diseñarla de manera correcta, sabiendo qué hacer y cómo hacerlo de la manera adecuada y más idónea.

El siguiente paso, obviamente, ha sido la creación de una ontología básica, que utilizaba una serie de datos limitados, que era capaz de tomar decisiones para situaciones muy concretas. En este punto no se han tenido en cuenta todavía las diferentes velocidades de los vehículos, sino que se suponía que todos ellos circulaban a una velocidad igual o similar. De esta manera se ha simplificado en gran medida la dificultad del problema, permitiendo así la realización de un sistema de decisiones básico, el cual se completaría de manera incremental en etapas posteriores.

Los datos que se han tenido en cuenta para la toma de decisiones han sido básicamente el carril y la posición de los vehículos en la carretera, y las soluciones aportadas eran limitadas, simples y generales, pero han supuesto el núcleo y la base de todo lo que ha sido añadido con posterioridad.

Más adelante se ha creado la primera versión de una aplicación prototipo que ha actuado como simulador, la cual ha utilizado la ontología creada en la fase anterior. Esta aplicación muestra de una manera mucho más visual y amigable para el usuario el proceso de recogida de información, la toma de decisiones y la aplicación de dichas soluciones al escenario para comprobar cómo se solventa la situación de emergencia.

Ésta ha sido muy trivial y simplemente consistía en posicionar los coches en la carretera y ver cómo se movían los mismos tras los procesos de toma de decisiones.

Después, se ha realizado otra etapa en la cual se ha tratado de incluir una mayor información y más funcionalidades a dicho prototipo. El aspecto principal que se ha añadido han sido las velocidades, que anteriormente no se tenían en cuenta. Por lo tanto, debido a esa modificación se han añadido nuevos elementos a la ontología, que además, han proporcionado nuevas soluciones y más detalladas.

En este punto de desarrollo, lo que se ha hecho ha sido adaptar el simulador ya realizado en fases anteriores a las nuevas modificaciones, de manera que se ha incrementado su funcionalidad, para adaptarla a su correspondiente especialización en la ontología a la que va ligada.

Se han añadido cuadros de texto, ligados a los vehículos, en los cuales el usuario debe introducir la velocidad a la que circulan los mismos. Además de esto, también se muestran por pantalla las acciones que realiza cada uno de ellos, viendo así las decisiones que se van tomando en cada momento y entendiendo el porqué de sus movimientos.

Después, se han realizado una serie de cambios en dicha aplicación, relacionados con la utilización de la ontología y el proceso de inferencia, buscando mejorar los tiempos obtenidos.

Se ha utilizado un método diferente a la hora de poblar la ontología de individuales y también al consultarla para extraer los resultados. También se ha utilizado un razonador diferente, el cual ha proporcionado un salto tremendamente grande en los tiempos necesarios para inferir los resultados con la ontología.

Finalmente, se ha creado un documento que engloba todo lo realizado, llamado ‘Ontology Based Road Traffic Management’, el cual ha sido aceptado en una conferencia en Calabria (Italia), a la cual se va a asistir para explicar el proyecto a los asistentes.

## CAPÍTULO 7: IMPLEMENTACIÓN

En este capítulo se detalla a gran nivel todas y cada una de las fases de desarrollo de software, las cuales han llevado a la versión actual del proyecto.

En el mismo se presentan las diferentes versiones realizadas de la ontología, así como también de la aplicación que actúa como simulador, mostrando el funcionamiento que tendría el sistema en un entorno real.

Finalmente, GPS y tecnologías inalámbricas han sido utilizadas en el sistema, y cada vehículo incluido en el mismo es un elemento inteligente, formando parte de un sistema distribuido.

El hecho de no tener un sistema centralizado significa que no es necesario tener un dispositivo extra para que éste actúe como un servidor, recolectando toda la información enviada por cada vehículo, tomando decisiones y transmitiéndolas hacia cada vehículo afectado en la carretera.

Java ha sido utilizado para el desarrollo de la aplicación que muestra el escenario y simula el proceso de toma de decisiones por parte de los vehículos afectados por la situación de emergencia.

Protégé ha sido la herramienta para almacenar la inteligencia en diferentes reglas que más tarde son comprobadas y ejecutadas para extraer las acciones que serán sugeridas a realizar a cada vehículo. Este conjunto de reglas, con una estructura concreta, es la ontología.

Los componentes más importantes de una ontología son las clases, las propiedades, los individuales y, sobre todo, las reglas SWRL:

- Las *clases* contienen la jerarquía del sistema.
- Las propiedades pueden ser asignadas tanto a las clases como a los datos.
  - Si son *propiedades de objeto*, lo que hacen es relacionar clases.
  - Si son *propiedades de datos*, aportan propiedades sobre las instancias de una clase.
- Los *individuales* son instancias de clases, y representan objetos en unos dominios definidos, que pertenecen a diferentes clases dependiendo de los valores asignados a sus propiedades.
- Las *reglas SWRL* son la inteligencia, las reglas que proporcionan los resultados finales dependiendo de la información aportada. Ellas describen los requisitos para que un individual pertenezca a una clase u otra, que se hará efectivo tras la inferencia.

La información que se aporta a la ontología para que haga sus inferencias está basada en la situación de la carretera en un momento determinado, que son las propiedades de cada vehículo afectado por la situación de emergencia.

El escenario estudiado en el proyecto es una carretera recta con dos carriles en la misma dirección y sentido, y las pautas principales que se han seguido a la hora de tomar decisiones en la ontología han sido las siguientes:

- Todos los vehículos son dirigidos hacia el carril más lento, intentando dejar paso libre al vehículo en situación prioritaria.
- En los diferentes carriles de la carretera se supone que, cuanto más a la izquierda está un vehículo, más rápido circula (aplicable solamente en la primera versión de la aplicación).
- Lo primero que se realiza es buscar un hueco lateral libre para cada vehículo que se tiene que incorporar al carril lento. Si no hay, se busca un hueco en una posición más adelantada. Si tampoco hay, se busca un hueco en una posición más retrasada. Si tampoco hay, lo que se hace finalmente es crear un hueco mediante el movimiento de los vehículos del carril lento.
- Una estrategia importante, que se ha establecido como norma general, ha sido la siguiente: 'acelerar para crear huecos libres y no molestar al vehículo en emergencia que se aproxima'.

## ONTOLOGÍA, PRIMERA VERSIÓN

La ontología creada de manera inicial ha sido bastante simple, con unas pocas reglas y un abanico limitado de resultados. Además, como ya se ha comentado anteriormente, se suponía que todos los vehículos circulaban a velocidades similares. Sin embargo, la última versión implementada ya contiene reglas más complejas y una variedad más amplia de acciones y resultados proporcionados. Ésta última versión es más realista, ya que puede trabajar con las diferentes velocidades de los vehículos.

Las *propiedades de datos* iniciales que se establecieron en la ontología fueron las siguientes:

- ✓ *isOvertaking*
- ✓ *isInFrontOfOvertaker*
- ✓ *hasSpaceLateral*
- ✓ *hasSpaceAhead*
- ✓ *hasSpaceBehind*

Basadas en esas propiedades, había varias *clases* en la ontología que representaban los diferentes resultados que se podían obtener, es decir, las acciones que se iban a sugerir a los conductores de los vehículos. Estas clases eran:

- ✓ *Accelerate*
- ✓ *Decelerate*
- ✓ *ChangeLane*
- ✓ *MaintainSpeed*

Las clases nombradas anteriormente tenían sus propias reglas especificadas, que establecían los requisitos que tenía que cumplir un individual (vehículo) para pertenecer a cada una de ellas. Todas esas *reglas* estaban basadas en el siguiente patrón:

***hasSpaceLateral = true***  
*overtaker: ChangeLine*  
*not overtakers: MaintainSpeed*

***hasSpaceLateral = false***  
***hasSpaceAhead = true***  
*overtaker: Accelerate*  
*not overtakers: MaintainSpeed*

***hasSpaceLateral = false***  
***hasSpaceAhead = false***  
***hasSpaceBehind = true***  
*overtaker: MaintainSpeed*  
*not overtakers:*  
    ***isInFrontOfOvertaker = true***  
        *MaintainSpeed*  
    ***isInFrontOfOvertater = false***  
        *Decelerate*



*hasSpaceLateral* = false  
*hasSpaceAhead* = false  
*hasSpaceBehind* = false  
 overtaker: Accelerate  
 not overtakers:  
     *isInFrontOfOvertaker* = true  
         Accelerate  
     *isInFrontOfOvertaker* = false  
         MaintainSpeed

## ONTOLOGÍA, VERSIÓN ACTUAL

La siguiente versión de la ontología es bastante diferente de todo lo explicado anteriormente. Las *propiedades de datos*, las *clases* y las *reglas* que almacenan la inteligencia contienen mucha más información y complejidad que anteriormente.

En este momento, las *propiedades de datos* de la ontología son las siguientes (Tabla 3), y se pueden ver también dentro de la aplicación Protégé en la figura que le sigue (Ilustración 12):

Ya incluidas en la versión anterior	Nuevas
<i>isOvertaking</i>	<i>beforeTheGap</i>
<i>isInFrontOfOvertaker</i>	<i>bigSpeedDifference</i>
<i>hasSpaceLateral</i>	<i>lastFirst-BeforeAfter-TheGap</i>
<i>hasSpaceAhead</i>	<i>speedWithAheadCar</i>
<i>hasSpaceBehind</i>	<i>speedWithBehindCar</i>
	<i>speedWithBeforeTheGapCar</i>
	<i>speedWithOvertaker</i>

Tabla 3: Propiedades de datos.

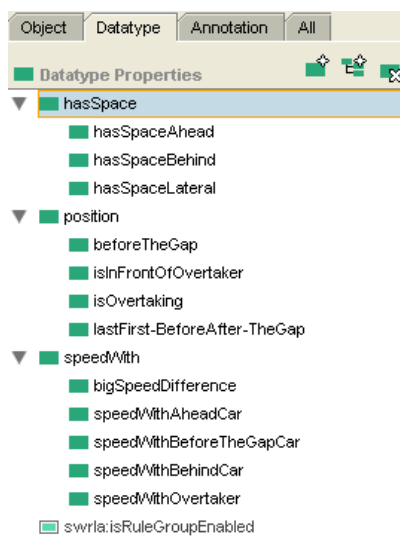


Ilustración 12: Propiedades de datos de la ontología.

Las últimas cuatro propiedades especificadas en la tabla anterior, relacionadas con la velocidad, tienen tres posibles valores:

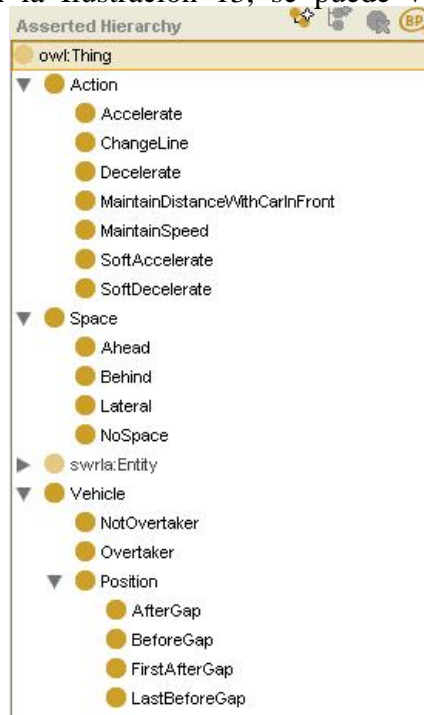
- ✓ *faster* (3)
- ✓ *same* (2)
- ✓ *slowlier* (1)

Las *clases* que contiene la ontología en esta nueva versión, las cuales están basadas en las propiedades anteriores, son:

Ya incluidas en la versión anterior	Nuevas
<i>Accelerate</i>	<i>MaintainDistanceWithCarInFront</i>
<i>Decelerate</i>	<i>SoftAccelerate</i>
<i>ChangeLane</i>	<i>SoftDecelerate</i>
<i>MaintainSpeed</i>	

**Tabla 4: Clases de la ontología.**

A continuación, en la Ilustración 13, se puede ver la jerarquía de clases que incluye las anteriormente citadas:



**Ilustración 13: Jerarquía de clases (ver Apéndice A).**

Dependiendo del valor booleano de la propiedad *bigSpeedDifference*, las acciones a realizar pueden variar entre *Decelerate* y *SoftDecelerate*, y entre *Accelerate* y *SoftAccelerate*.

Las reglas (ver Apéndice D), las cuales establecen a qué clase pertenece cada individual, están basadas en diferentes escenarios básicos, relacionados con la posición de los huecos a ocupar por los vehículos que tienen que incorporarse al carril más lento, para así dejar paso libre al vehículo en emergencia. Estos escenarios son presentados a continuación:

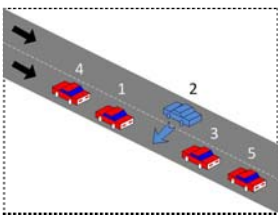
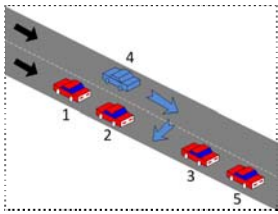
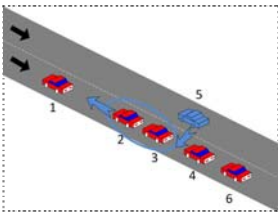
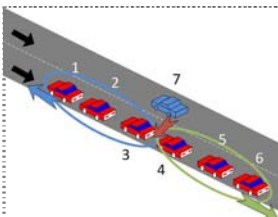
<i>HUECO LATERAL</i>	
	<b>1:</b> <i>MaintainDistanceWithCarInFront</i> <b>2:</b> <i>ChangeLine</i> <b>3:</b> <i>MaintainSpeed</i> <b>5 and all ahead:</b> Speed $\geq 1$ <i>Accelerate or MaintainSpeed</i> <b>4 and all behind:</b> Speed $\leq 1$ <i>Decelerate or MaintainSpeed</i>
<i>NO HUECO LATERAL, HUECO DELANTE</i>	
	<b>1 and all behind:</b> Speed $\leq 2$ <i>Decelerate or MaintainSpeed</i> <b>2:</b> <i>MaintainDistanceWithCarInFront</i> <b>3:</b> <i>MaintainSpeed</i> <b>5 and all ahead:</b> Speed $\geq 2$ <i>Accelerate or MaintainSpeed</i> <b>4:</b> <i>Accelerate</i>
<i>NO HUECO LATERAL, NO HUECO DELANTE, HUECO DETRÁS</i>	
	<b>1 and all behind:</b> Speed $< 5$ <i>Decelerate or MaintainSpeed</i> <b>2 and 3:</b> <i>Decelerate</i> <b>4 and all ahead:</b> Speed $> 3$ <i>Accelerate or MaintainSpeed</i> <b>5:</b> Speed $\leq 2$ and $> 3$ <i>Accelerate or Decelerate or MaintainSpeed</i>
<i>NO HUECO LATERAL, NO HUECO DELANTE, NO HUECO DETRÁS</i>	
	<b>1 and all behind:</b> <i>SoftDecelerate</i> <b>4 and all ahead:</b> <i>SoftAccelerate</i> <b>7:</b> Speed $> 3$ and $\leq 4$ <i>Accelerate or Decelerate or MaintainSpeed</i>

Tabla 5: Escenarios de emergencia, y correspondientes sugerencias por parte de la ontología. Vehículos numerados del 1 al 7 (parte izquierda); y sugerencias de la ontología (parte derecha).

El último escenario presentado en la tabla anterior podría ser el peor que se podría presentar, en el cual el vehículo número 7 necesita encontrar un hueco libre en el carril más lento, pero no hay ninguno. Entonces, los demás vehículos tienen que actuar en consecuencia para crear uno, donde el vehículo 7 pueda incorporarse.

Así, los vehículos del 1 al 3 y todos los de detrás de ellos (si hay alguno) recibirán una sugerencia para hacer *SoftDecelerate*, mientras que los vehículos del 4 al 6 y todos los de delante de ellos (si hay alguno) recibirán la sugerencia de *SoftAccelerate*. De esta manera, un nuevo hueco

será creado y el vehículo 7 deberá mantener su velocidad entre la velocidad del vehículo 3 y la del 4 mediante *Accelerate*, *Decelerate* o *MaintainSpeed* para poder incorporarse en breve.

Si todo se realiza correctamente y como se sugiere por parte de la ontología, el escenario se transformará en el primero mostrado en la tabla anterior (Tabla 5), donde el vehículo 7 tendrá un hueco libre lateral y con un simple *ChangeLane* la situación será completamente resuelta.

Como el sistema es dinámico, cada uno de los movimientos realizados por los vehículos es detectado y compartido instantáneamente. Así, si un conductor actúa de una manera diferente a la esperada, esto será detectado y se adaptarán las sugerencias a la actual situación.

A la hora de realizar el diseño de estas decisiones, se ha tenido que emplear mucho tiempo en razonar las estrategias a utilizar para guiar a los vehículos afectados por la situación de emergencia. Los patrones que marcan las acciones a realizar se pueden ver en el Apéndice E.

Durante este proceso se llegó a la conclusión de que siempre va a haber inconsistencia en la toma de decisiones en los casos en los que haya más de un vehículo que se tenga que incorporar desde el carril rápido al más lento. Esto es debido a que, al intentar solucionar todas las incorporaciones al mismo tiempo, unos vehículos tienen que reaccionar de una manera para permitir que un adelantador se incorpore, pero a su vez tienen que realizar otros movimientos diferentes para que se incorpore otro adelantador, y así sucesivamente por cada uno de los vehículos que se encuentran en el carril más rápido.

Para encontrar una solución a este problema se ha aceptado finalmente que prácticamente ningún caso va a poder ser solucionado en un paso, es decir la situación de emergencia se va a ir solucionando de manera progresiva. Por lo tanto, en caso de encontrarse con esta situación de inconsistencia de acciones a realizar por un mismo vehículo, se ha decidido que el mismo tendrá que realizar la acción que solventa el área más cercana al vehículo de emergencia, permitiendo así ir limpiando el carril rápido progresivamente.

Lo más inmediato al vehículo prioritario es para lo que más urge encontrar una solución, así se irá permitiendo el avance del mismo, interfiriendo en su camino lo más mínimo posible.

La ontología final creada está basada en setenta y siete reglas, como se muestra en la siguiente tabla (Tabla 6), y la estructura que siguen todas ellas es:

*Condición1, Condición2 ... CondiciónN → Resultado*

Elemento	Cantidad
Clases	24
Propiedades	12
Reglas	77

**Tabla 6: Elementos de la ontología.**

En esta última versión también se cambiaron las reglas, eliminando la creación de individuales intermedios, los cuales ralentizaban el proceso de inferencia. Esto también dificultaba más tarde la extracción de los resultados en el simulador, ya que había más individuales que pertenecían a más

clases a su vez, lo que provocaba la necesidad de filtrar los resultados para obtener las acciones a sugerir al conductor.

## ***SIMULADOR, PRIMERA VERSIÓN***

Lo primero que se realizó fue la implementación del interfaz de usuario de la aplicación. Era bastante simple, solamente contenía una carretera con dos carriles y un botón para iniciar la simulación. En la carretera, un vehículo de emergencia (ambulancia en este caso) ya estaba dibujado, y necesitaba tener paso libre para circular a lo largo de la misma lo más rápido posible.

El proceso seguido por la aplicación para simular el flujo del tráfico, teniendo en cuenta las acciones inferidas por la ontología, se describe a continuación:

- El usuario del simulador tiene que posicionar los vehículos mediante clics sobre la carretera en las posiciones exactas en las que serán dispuestos. Cuando se pincha con el ratón sobre un punto de la carretera, un vehículo es automáticamente situado en la posición en la que se encuentra el ratón.
- En este momento, la variable *isOvertaking* se establece a *true* o *false*. Si el vehículo situado se encuentra en el carril rápido (el de arriba en la aplicación) su valor es *true*; si por el contrario se posiciona en el carril lento (el de abajo), el valor es establecido como *false*.
- Después de haber situado todos los vehículos necesarios para la simulación (a elección del usuario), las posiciones entre los vehículos de un carril y del otro son revisadas para establecer el valor *isInFrontOfOvertaker* de cada vehículo del carril lento a *true* o a *false*.
- Ahora, el siguiente paso es buscar huecos libres en el carril lento y asignarlos a los vehículos:
  - Primero, un hueco lateral se intenta asignar a cada vehículo.
  - Más tarde, para cada vehículo que no tiene todavía un hueco asignado, se busca uno en una posición más adelantada con respecto al mismo.
  - Después, un hueco es buscado para los vehículos que todavía no tienen ninguno asignado. Ahora se busca el hueco en posiciones más retrasadas al vehículo concreto.
  - Finalmente, para todos aquellos vehículos que no se ha encontrado ningún hueco ni lateral, ni delante, ni detrás, se creará uno nuevo mediante la modificación de las velocidades de los que se encuentran en el carril lento.

Si un hueco en concreto puede ser asignado a más de un vehículo, el seleccionado es el que se encuentra más cercano al mismo, siempre teniendo en cuenta la inercia del tráfico. Es decir, cuando se busque un hueco delante para un vehículo, el que tendrá preferencia será el que se encuentre más cercano al mismo (en una posición más adelantada); y cuando se busque un hueco detrás, tendrá preferencia el vehículo más retrasado, ya que está más cercano al hueco.

Para evitar asignar un mismo hueco a más de un vehículo, se utiliza un proceso de reserva en la aplicación, de manera que se almacena toda la información sobre esas asignaciones para llevar un control y evitar problemas que pudieran surgir.

Se ha establecido una longitud mínima y máxima de hueco. Cuando uno es muy largo, éste se divide en varios, cada uno de ellos con una longitud concreta. Puede haber unos huecos más grandes que otros, pero nunca llegarán a ser dos veces la longitud de los otros, ya que si fuera así podrían ser divididos en dos individuales.

Esta es la razón por la que los huecos a asignar pueden ser modificados en tamaño y posición, para aprovecharlos de la mejor manera posible a la hora de asignarlos a los distintos vehículos.

En este punto, la única tarea que quedaba por realizar para que todo funcionase era el uso de la ontología creada en pasos anteriores de la implementación del proyecto.

Para poder acceder desde la aplicación Java al fichero de la ontología e interpretar los datos escritos en él, era necesario utilizar unas librerías concretas (entre todas ellas, 'Jena' es la más importante). Ellas proporcionaban todas las funcionalidades requeridas para trabajar con la ontología creada.

El proceso que seguía la aplicación para utilizar dicha ontología se presenta a continuación:

- Para cada vehículo que adelanta (carril rápido) en el escenario, hacer:
  - o Crear el vehículo como un individual, con sus propiedades de datos.
  - o Crear los vehículos que no adelantan como individuales, estableciendo también sus propiedades de datos.

Estas propiedades fueron almacenadas en variables en el momento que se posicionaron los vehículos en la carretera.

A continuación se muestran extractos de código en los que se realiza este trabajo:

```
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX table:
<http://www.traffic.com/Ontologies/traffic2.owl#>
INSERT DATA{
    table:Car c2.id rdf:type table:Vehicle .
    table:Car c2.id table:hasSpaceLateral c1.hasSpaceLateral ;
    table:hasSpaceAhead c1.hasSpaceAhead ;
    table:hasSpaceBehind c1.hasSpaceBehind ;
    table:isInFrontOfOvertaker ov.isInFrontOfOvertaker ;
    table:isOvertaking false;
    table:beforeTheGap c2.beforeTheGap ;
    table:lastFirst-BeforeAfter-TheGap false;
    table:speedWithBeforeTheGapCar " +
c2.speedWithBeforeTheGapCar ;
    table:bigSpeedDifference c2.bigSpeedDifference ;
}
```

```

UpdateRequest query = UpdateFactory.create(queryString);

// Execute the query and obtain the results
UpdateAction.execute(query, model);

// Save the result to file
FileWriter out = new FileWriter("C:/.../traffic2.owl");
model.write(out,null,"RDF/XML-ABBREV");

```

**Ilustración 15: Ejecución de la consulta SPARQL.**

- Ejecutar la ontología y extraer los resultados.  
Estos resultados son las acciones inferidas, las cuales se sugieren a los conductores para que las realicen. Un extracto del código se muestra en la siguiente figura:

```

public static void extractActions(Car c1) {
try {
    File f = new File( "C:/.../traffic2.owl");
    FileInputStream fin = new FileInputStream(f);

    // Load the ontology from the specified URL
    OWLModel owlModel =
ProtegeOWL.createJenaOWLModelFromInputStream(fin);
    //runs example with the Pellet reasoner (accessed through Jena)
    run(owlModel, createPelletJenaReasoner(owlModel), c1);

    } catch (Exception e) {
        e.printStackTrace();
    }
}

```

**Ilustración 16: Extraer acciones.**

- Almacenar estos resultados obtenidos (acciones).
- Eliminar todos los vehículos (individuales) de la ontología para así tenerla limpia y preparada para el siguiente uso.

Después de realizar todo este proceso de manera completa, era tiempo de comprobar las diferentes acciones que tiene que llevar a cabo cada vehículo. Si hay alguno que tenía que realizar más de una diferente, había incoherencia porque es imposible que un vehículo pueda reaccionar de dos maneras distintas al mismo tiempo. Por lo tanto, la mejor solución era que se realizase la que va relacionada con el vehículo que adelanta que está más cercano al vehículo de emergencia. Como ya se ha comentado anteriormente, con esta estrategia se solucionan los problemas más inmediatos en el camino del vehículo prioritario, despejando el área más cercana al mismo. Incluso, actuando de este modo, podrían solucionarse también otros problemas que habría que encarar en los siguientes pasos.

Las funciones principales que describen el proceso completo, partiendo del clic en el botón ‘Start’ del simulador, se muestran a continuación:

```

startBut.addMouseListener(new
java.awt.event.MouseAdapter() {
    @Override
    public void mousePressed(MouseEvent e)
    {
        new Thread(new Runnable() {
            @Override
            public void run() {
                loop();
            }
        }).start();
    }
});

```

**Ilustración 18: Vehículo posicionado sobre la carretera.**

```

public static void loop() {
    while (true) {
        try {
            moving = true;

            noMoreCars();
            ontology();
            if (end) {
                break;
            }
            Thread.sleep(500);
            move();
            canv.repaint();
        } catch (InterruptedException ex) {
            System.out.println("Error loop" + ex.toString());
        }
    }
}

```

**Ilustración 17: Estructura del simulador.**

Cabe decir que para poder manejar la ontología desde la aplicación en Java se encontraron bastantes problemas. Se estuvo bastante tiempo buscando la solución al problema de referencias que hacía que las funciones utilizadas no trabajasen de manera correcta. Se estuvieron probando diferentes versiones de Protégé, así como también distintas versiones de librerías extraídas directamente de los repositorios oficiales (<http://smi-protege.stanford.edu/repos/protege/>), pero no se conseguía hacerlo funcionar.

Finalmente, la solución estuvo en importar directamente a la aplicación Java las librerías de Protégé. Éstas son las que utiliza el simulador y las que permiten que todo funcione perfectamente y sin ningún tipo de fallo.

## ***SIMULADOR, VERSIÓN ACTUAL***

La última versión desarrollada de la aplicación es bastante similar a la anterior, con respecto a la interfaz de usuario. Sin embargo, hay varias diferencias básicas que la hacen mucho más precisa en los movimientos realizados por los vehículos:

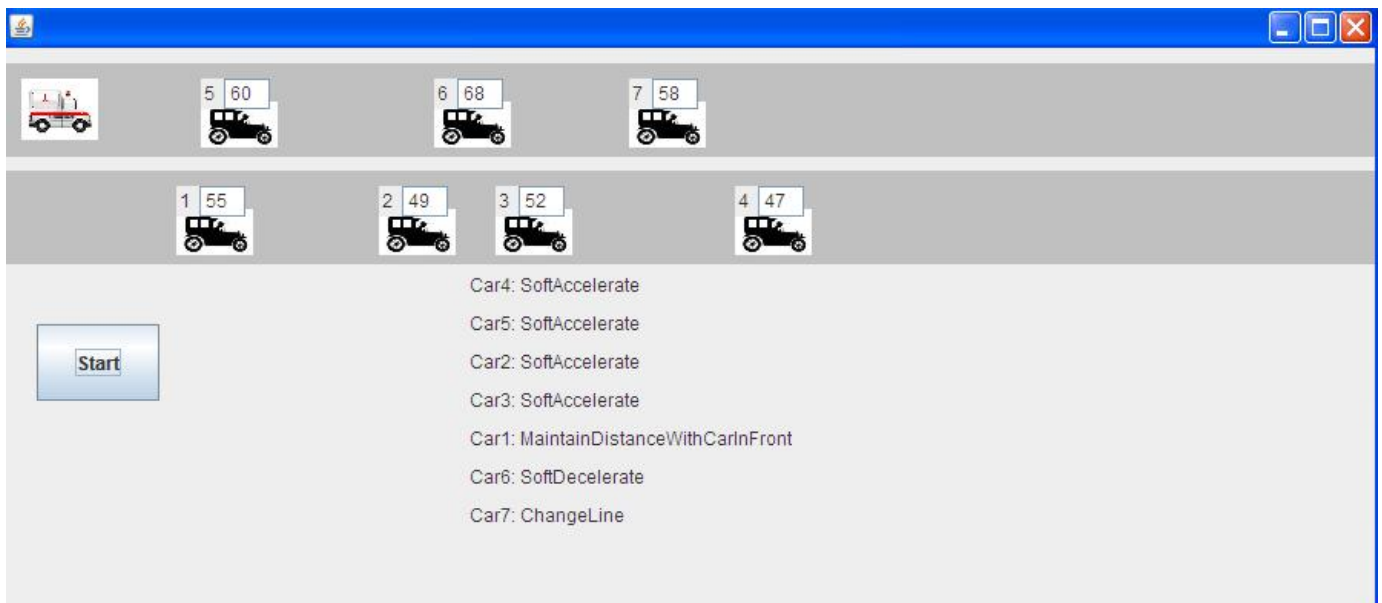
- La diferencia principal sucede en el momento de posicionar los vehículos sobre la carretera: cuando se hace clic, un área de texto aparece con el cursor en él pidiendo al usuario que especifique la velocidad a la que circula ese concreto vehículo.
- Las acciones a realizar por esos vehículos ahora son mostradas en la aplicación, de manera que se proporciona información más detallada sobre los procesos internos y decisiones que se toman a la hora de distribuir el tráfico.



- A lo largo de la simulación, las velocidades y los movimientos de cada vehículo son modificados, siempre basados en y relacionados con las velocidades especificadas por el usuario en los primeros pasos.

Estas actualizaciones hacen a la aplicación estar mucho más cerca del comportamiento real de todos los eventos que se suceden en la carretera, basando sus movimientos en las velocidades de los vehículos que circulan sobre ella, así como en sus posiciones (la versión anterior solamente se basaba en esto último, en las posiciones).

En la siguiente figura (Ilustración 19) se presenta una captura de pantalla de la aplicación, durante una simulación de una situación de emergencia. En ella se puede percibir cuál es la situación y cuáles son los movimientos que se sugieren al conductor de cada vehículo. En la simulación se realizan directamente esas acciones sugeridas.



**Ilustración 19: Simulador desarrollado.**

Además de los cambios en la apariencia de la misma, también se han realizado modificaciones internas, referentes a la programación del simulador:

- Una de ellas ha sido el cambio de la manera de interactuar con la ontología. Anteriormente se utilizaban consultas SPARQL para realizar inserciones, eliminaciones y consultas sobre la misma. Ahora no se utiliza esa metodología, sino que se hace uso de otras funciones que provee la API, de manera que se va haciendo todo mediante la creación de instancias y sin escribir texto de consulta.

```
if (c1.hasSpaceAhead == 1){
    OWLIndividual car =
owlModel.getOWLNamedClass("Vehicle").createOWLIndividual("Car"+c1.id);
    car.setPropertyValue(owlModel.getRDFProperty("hasSpaceLateral"), c1.hasSpaceLateral);
    car.setPropertyValue(owlModel.getRDFProperty("hasSpaceAhead"), c1.hasSpaceAhead);
    car.setPropertyValue(owlModel.getRDFProperty("hasSpaceBehind"), c1.hasSpaceBehind);
    car.setPropertyValue(owlModel.getRDFProperty("isInFrontOfOvertaker"), 0);
    car.setPropertyValue(owlModel.getRDFProperty("isOvertaking"), 1);
    car.setPropertyValue(owlModel.getRDFProperty("speedWithBeforeTheGapCar"),
    c1.speedWithBeforeTheGapCar);
    car.setPropertyValue(owlModel.getRDFProperty("hisSpeedDifference"), c1.hisSpeedDifference);
}
```

**Ilustración 20: Nueva manera de insertar individuales.**

Esto ha supuesto una pequeña mejora, aunque no muy significativa, en los tiempos obtenidos para realizar las inferencias y obtener los resultados. La comparativa de los dos métodos se muestra en la siguiente tabla (Tabla 7):

		SPARQL (ms)	INSTANCIAS (ms)
<b>1 Vehículo adelantando</b>	<b>Prueba 1</b>	9985	9765
	<b>Prueba 2</b>	9812	9719
	<b>Prueba 3</b>	9687	9453
<b>2 Vehículos adelantando</b>	<b>Prueba 1</b>	9547 + 9156	9516 + 9171
	<b>Prueba 2</b>	9656 + 9234	9547 + 9172
<b>3 Vehículos adelantando</b>	<b>Prueba 1</b>	9937 + 9140 + 9141	9672 + 9281 + 9109
	<b>Prueba 2</b>	12484 + 11984 + 11938	9625 + 9188 + 9141

Tabla 7: Comparativa de tiempos obtenidos al inferir.

- Otra mejora incorporada, y de hecho la más importante de todo el proyecto, ha sido el cambio de razonador utilizado para realizar las inferencias. Anteriormente se utilizaba el razonador Pellet, pero finalmente se ha optado por la utilización del razonador Jess ([15]).

```
//Creamos el puente entre JESS y la ontología
SWRLRuleEngine ruleEngine = SWRLRuleEngineFactory.create(owlModel);

//Ejecutamos la inferencia
ruleEngine.infer();
```

Ilustración 21: Inferencia con Jess.

Los tiempos se han visto reducidos drásticamente, pasando de más de 9 segundos a alrededor de 300 milisegundos (ver tabla 8). Esto demuestra que el sistema desarrollado en el proyecto puede operar en tiempo real, ofreciendo soluciones inmediatas a peligros surgidos en la carretera.

Vehículos	# Total	Tiempo/inferencia
-----------	---------	-------------------

(Adelantando)	inferencias	(ms)
5 (1)	1	359
6 (1)	1	343
6 (2)	3	280
7 (2)	3	280
9 (4)	8	242

**Tabla 8: Comparativa de tiempos obtenidos al inferir.**

Como se puede apreciar en la tabla anterior, conforme el sistema va progresando, el grafo de la ontología es evaluado con mayor eficiencia, traduciéndose en inferencias más rápidas.

- También se ha tomado otra decisión que ha afectado a la velocidad de procesamiento, ya no directamente a la inferencia, sino a la computación de datos por parte del simulador. Se ha decidido cargar la ontología en la aplicación una sola vez y a partir de ahí ir trabajando sobre ella, sin ir reescribiendo sobre el fichero. De esta manera se obtienen muchas ventajas:
  - ✓ Ahorro de tiempo al cargar la ontología solamente una vez.
  - ✓ Ahorro de tiempo al no escribir sobre el fichero.
  - ✓ Ahorro de tiempo al no tener que limpiar el fichero en cada iteración para volver a cargarlo.
  - ✓ Imposible corromper la ontología, ya que no se va a modificar nada sobre ella directamente en el fichero.
  - ✓ Al no escribir sobre el mismo, no hay que controlar cuándo y cómo se cierra.
- Para agilizar la computación por parte del simulador, también se ha realizado un cambio bastante importante. Anteriormente, cada vez que se realizaba un mínimo movimiento por cada coche se volvía a realizar la inferencia, incluso si el escenario estaba exactamente igual que anteriormente. Esto suponía una pérdida de tiempo inaceptable, por lo que se solucionó controlando los movimientos de los vehículos. De esta manera se controla cuándo se crean, destruyen y asignan huecos, y cuándo un vehículo está en paralelo a un hueco que tiene que ocupar. Estos momentos son los que marcan que hay que realizar una nueva inferencia, porque el escenario ha cambiado significativamente y se necesita asignar nuevas acciones a los vehículos.
- Se han encontrado problemas durante la simulación: ocasionalmente, algunos vehículos colisionaban. Para solucionarlo, se ha hecho que los vehículos avancen proporcionalmente a las velocidades y con respecto a las acciones que tienen que realizar. Una situación especialmente problemática se producía cuando un vehículo tenía que hacer *MaintainDistanceWithCarInFront*. Eso supuso que había que tener en cuenta, además de la diferencia de velocidades entre vehículos, la acción a realizar por parte del vehículo que se encontraba delante del mismo.

## CAPÍTULO 8: PRUEBAS

En este capítulo se aporta una explicación de los momentos en los que se han realizado las pruebas al software desarrollado, y la razón por la que han sido elegidos esos momentos en concreto.

También se muestran unos ejemplos de pruebas de caja negra, pruebas de caja blanca y pruebas de carga que han sido aplicadas al proyecto realizado, asegurando así el correcto funcionamiento del mismo.

A lo largo de todas las etapas del proyecto se han ido realizando pruebas, al mismo tiempo que se desarrollaba todo el proceso de creación de software. Estas pruebas no han sido planificadas ni se ha definido ningún calendario en el cual se establezcan los momentos temporales exactos en los cuales hay que hacer unas pruebas concretas.

Lo que ha determinado los momentos exactos en los cuales eran necesarias dichas pruebas han sido las distintas fases de desarrollo de software, definidas en el capítulo ‘Metodología’ de este documento. En el momento que se terminaba de desarrollar una fase, se realizaban una serie de pruebas, tanto de caja negra como de caja blanca, que eran las que marcaban el momento de pasar a la siguiente fase.

Si dichas pruebas mostraban unos resultados erróneos o inesperados, se seguía la siguiente metodología:

- 1) Se revisaba lo que se estaba probando para localizar el error.
- 2) En el momento en el que se encontraba, se reparaba.
- 3) Se volvía a realizar la misma prueba, con los mismos datos.

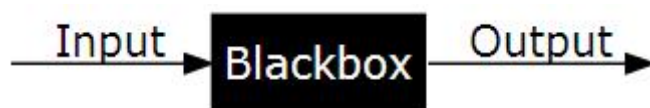
Cuando se obtenía el resultado esperado, se volvía a realizar la misma prueba, pero con diferentes datos. Si se encontraba algún fallo, se volvían a hacer todos los pasos. Finalmente, en el momento en el que ya se había realizado la misma prueba con diferentes datos, obteniendo resultados satisfactorios, era el momento de pasar a la siguiente prueba, siguiendo la misma metodología que con todas las anteriores.

Así, cuando ya se habían realizado todas las pruebas requeridas y se había obtenido el cien por ciento de resultados correctos, se pasaba a la siguiente fase de desarrollo del proyecto, asegurando así que todo lo realizado anteriormente cumplía las expectativas y los requisitos necesarios para continuar.

La elección de esos momentos en los que se realizaban las pruebas, está basada principalmente en el sentido común, ya que es obvio que hay que hacer pruebas al terminar una fase, con anterioridad al inicio de la siguiente. Si no se hace así, posibles errores y malfuncionamientos se pueden ir arrastrando y expandiendo por todo el sistema, de manera que más adelante sea una tarea imposible la reparación, y una tarea muy costosa el simple hecho de encontrarlos, seguida de sus consiguientes reparaciones, las cuales supondrían un mayor esfuerzo que se si se hubieran realizado con anterioridad.

## **CAJA NEGRA**

Varias de esas pruebas que se han llevado a cabo han sido de caja negra, es decir, se han aportado una serie de datos de entrada al sistema, y se han obtenido una serie de salidas a partir de los mismos. Todo este proceso ha sido realizado sin tener en cuenta el funcionamiento interno.



**Ilustración 22: Caja negra.**

Una de esas pruebas de caja negra realizadas se muestra a continuación, en la Tabla 9:

ENTRADA	SALIDA ESPERADA
<b>Vehículo 1:</b> carril lento, 45 km/h  <b>Vehículo 2:</b> carril lento, 47 km/h, delante de vehículo 1  <b>Vehículo 3:</b> carril rápido, 50 km/h, espacio para incorporarse entre vehículos 1 y 2, pero retrasado con respecto al mismo	<b>Vehículo 1:</b> MaintainDistanceWithCarInFront <b>Vehículo 2:</b> MaintainSpeed <b>Vehículo 3:</b> SoftAccelerate
‘Salida esperada’ del paso anterior	<b>Vehículo 1:</b> MaintainDistanceWithCarInFront, velocidad igual a la del vehículo 2  <b>Vehículo 2:</b> MaintainSpeed <b>Vehículo 3:</b> ChangeLane

**Tabla 9: Prueba de caja negra.**

Con referencia a la tabla anterior, partiendo de las entradas, lo que se esperan como salidas son muestras de cómo va a ir evolucionando el tráfico para poder solucionarse la situación de emergencia.

Al haber un hueco libre entre los vehículos 1 y 2, éstos van a adaptar sus velocidades para que ese espacio se mantenga, para que el vehículo 3 se incorpore al mismo. De esta manera, cuando el vehículo 3 se encuentre paralelo a dicho hueco, lo que tendrá que hacer es un simple movimiento lateral para cambiar de carril, dejando así paso libre para el vehículo que tiene paso prioritario por el carril más rápido.

## CAJA BLANCA

Además de estas pruebas de caja negra, también se han realizado diversas pruebas de caja blanca, en las cuales nos hemos centrado en los detalles procedimentales del software, comprobando y asegurando el correcto funcionamiento de la programación del código fuente.

Se han realizado estas pruebas con diferentes valores de entrada, examinado así todos los posibles flujos de ejecución del software, cerciorándonos de que se devuelven los valores de salida adecuados.

Un ejemplo de esas pruebas de caja blanca se ha realizado sobre la parte del código que se encarga de asignar huecos libres en el carril lento a los vehículos que se encuentran en el rápido, y necesitan cambiar de carril para facilitar el paso del vehículo en emergencia.

El extracto del mismo se muestra a continuación, en la Ilustración 23:

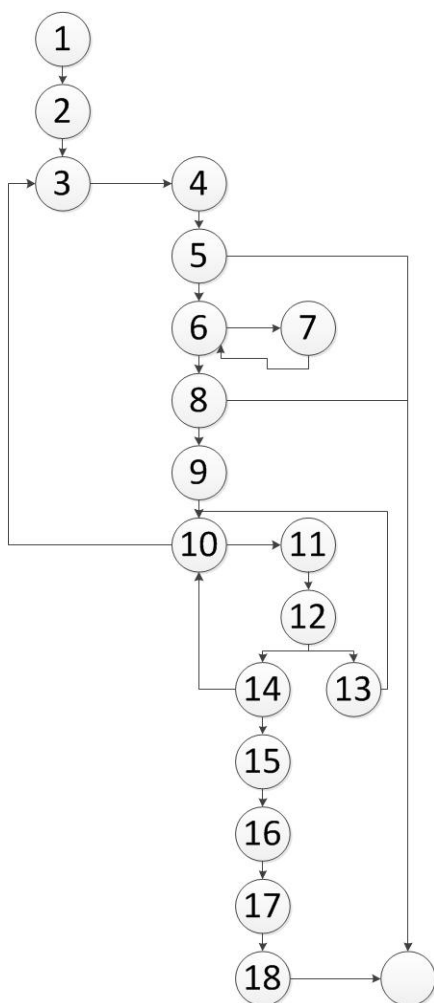
```

1 Collections.reverse(cars);
2 Iterator it6 = cars.iterator();
3 while (it6.hasNext()){
4     Car c = (Car)it6.next();
5     if (!it6.hasNext()){ break;}
6     while (((c.isOvertaking == 0) || ((c.isOvertaking == 1) && (c.hasSpaceLateral == 1))) &&
7         it6.hasNext()){
8         c = (Car)it6.next();
9     }
10    if ((c.isOvertaking == 0) || ((c.isOvertaking == 1) && (c.hasSpaceLateral == 1))){break;}
11    Iterator it5 = huecos.iterator();
12    while (it5.hasNext()){
13        Hueco h = (Hueco)it5.next();
14        if (h.occupied){
15            System.out.println("occupiedAhead: " + h.occupied + "; h.start=" + h.start + " ;h.end="
16                + h.end);
17        }
18        else{
19            if ((c.position.x)<h.start){
20                System.out.println("Hueco ahead");
21                asignarHueco(h,c,false,true,false);
22                notOvertakersProperties(h, "ahead");
23                compareSpeedsWithBeforeTheGapCar(c);
24                break;
25            }
26        }
27    }
28 }

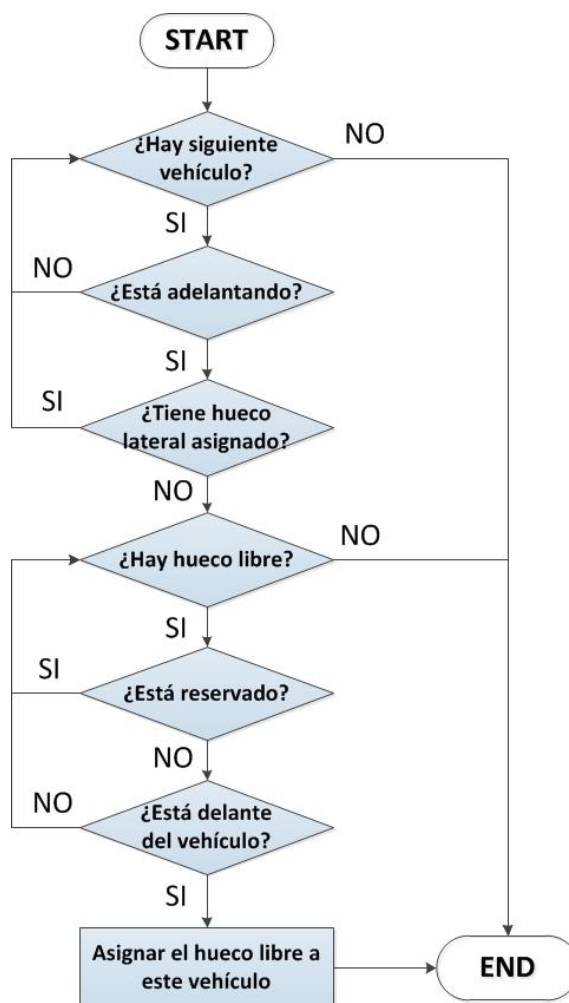
```

**Ilustración 23: Extracto de código que asigna huecos adelantados.**

Partiendo del mismo, a continuación se presenta el diagrama que muestra todos los caminos que se pueden seguir (Ilustración 24). También se muestra otro gráfico más simplificado y comprensible del mismo fragmento de código (Ilustración 25):



**Ilustración 24: Prueba de caja blanca.**



**Ilustración 25: Prueba de caja blanca (simplificada).**

## PRUEBAS DE CARGA

Este tipo de pruebas se han realizado para observar el comportamiento del software bajo una cantidad de peticiones esperada. Adaptado a nuestro proyecto, se han realizado sobre el simulador, estableciendo un número de vehículos que será factible que se encuentren en situaciones reales.

Un ejemplo de estas pruebas de carga se ha mostrado anteriormente en el capítulo ‘Implementación’, en el sub apartado ‘Simulador, versión actual’. En el mismo se puede ver la siguiente tabla (Tabla 10) con una serie de pruebas que se han realizado sobre el simulador. Estas pruebas se asemejan a los escenarios con los que tendrá que trabajar el sistema implantado en situaciones reales.



<b>Vehículos (Overtakers)</b>	<b># Total inferencias</b>	<b>Tiempo/inferencia (ms)</b>
5 (1)	1	359
6 (1)	1	343
6 (2)	3	280
7 (2)	3	280
9 (4)	8	242

**Tabla 10: Comparativa de tiempos obtenidos al inferir.**

Con la obtención de estos resultados, es posible determinar que el sistema desarrollado soporta perfectamente la carga que será requerida. Esto es debido a que su funcionamiento no se ha visto interrumpido por ningún problema durante la prueba, y los tiempos de inferencia son perfectamente aplicables al entorno real.

## CAPÍTULO 9: DISCUSIÓN

En este capítulo se realiza un proceso de razonamiento en el cual se exponen los puntos más controvertidos del proyecto desarrollado. Se explican las razones que han llevado a las elecciones tomadas a la hora de establecer unas pautas que marcan e identifican al proyecto.

El proyecto desarrollado tiene una serie de diferencias con respecto a lo que realizado hasta ahora en otros proyectos, es lo que le da individualidad y lo caracteriza. Todas estas decisiones tomadas para establecer las bases del mismo y lo que lo caracteriza no ha sido un proceso trivial, ha sido cuidadosamente estudiado y todas las decisiones tomadas han sido estudiadas y comparadas con las demás opciones disponibles.

Todo esto nos ha hecho razonar profundamente sobre una serie de aspectos concretos sobre el proyecto, como por ejemplo la estructura distribuida del mismo, en el cual cada vehículo contiene inteligencia y toma sus propias decisiones.

Si, por el contrario, se tratara de un sistema centralizado y hubiera un punto central que actuase como servidor de datos, aparecería un nuevo problema a afrontar en el proyecto: ¿Dónde está situado este nodo central? ¿Está instalado en un vehículo? ¿Está situado en una instalación dentro de un edificio? Esto supondría que se necesitaría cierto tipo de infraestructura y, como consecuencia, un mayor presupuesto en el proyecto, lo que sería un importante inconveniente ya que el presupuesto es siempre el mayor hándicap que limita todo lo que se realiza dentro de un proyecto.

Hay muchas cuestiones a las que habría que responder en caso de elegir un sistema centralizado, además de no tener grandes ventajas con respecto a un sistema distribuido. De hecho, para el proyecto actual no habría ninguna o casi ninguna ventaja.

El proyecto actual se extiende a espacios más reducidos que otros proyectos ya desarrollados (no grandes áreas metropolitanas de muchas carreteras) y focaliza su atención en la distribución de los vehículos para facilitar, de manera segura, el camino del vehículo en situación de emergencia, intentando modificar en la menor medida de lo posible el escenario. También, las acciones sugeridas a cada vehículo son específicas de cada uno de ellos, no básicas, y están relacionadas con la velocidad, la posición y las distancias que existen entre ellos.

Algunos estudios han demostrado que el factor más determinante para que se produzca tráfico denso y atascos son los errores humanos [14]. Esto supone que, incluso informando a los conductores sobre las acciones que deberían realizar, no se asegura de manera definitiva que éstos reaccionen de la manera deseada.

Hay también dos proyectos, desarrollados por Google y la Universidad Alemana de Freie, que están construyendo y probando un vehículo que es capaz de conducirse por sí solo, sin la interacción de un conductor humano. En este punto es necesario razonar si la sociedad está preparada para aceptar estas situaciones, y también saber si en algún momento en el futuro será seguro dejar a un aparato electrónico el control completo del vehículo, lo que podría dejar en sus manos la integridad física de los ocupantes del mismo. Los conductores quieren sentir que tienen el control del vehículo, y no en el sentido contrario, dejando que decida y actúe el último.



**Ilustración 26: Vehículo conducido sin interacción humana.**

Cuando se ha desarrollado la primera implementación del simulador, algunas inconsistencias han sido encontradas, como ya se ha comentado anteriormente, cuando se necesitaba que un mismo vehículo realizase dos movimientos distintos al mismo tiempo. Esto es debido al hecho de que se intenta facilitar los movimientos de todos los vehículos que se encuentran en el carril más rápido, y así las instrucciones pueden ser diferentes para cada uno de ellos.

Esto no es posible de realizar físicamente, por lo que la solución ha sido elegir las acciones a realizar que facilitan los movimientos del vehículo rápido que se encuentra más cercano al vehículo al que hay que dejar paso libre. De esta manera, se va solucionando siempre primero el problema más inmediato y, quizás, con estas acciones otros problemas futuros también son solventados.

Todos los demás trabajos sobre el tema utilizan GPS como tecnología para obtener la posición de los vehículos, tal y como está planteado que sea en el presente proyecto. El punto diferenciador y crítico está en que en este proyecto, además del GPS se tiene previsto utilizar una tecnología extra para asegurar que las posiciones de los vehículos son exactas, ya que solamente con el primero se puede tener un rango de error desde 25 hasta 100 metros. Con un error de posicionamiento de sólo 3 metros podría haber muchos problemas en las tomas de decisiones, ya que en el proyecto se trabaja con distancias cortas, por lo que es necesario esa incorporación adicional de otra tecnología adicional.

Finalmente, algunos de otros proyectos ya realizados sugieren el hecho de que no todos los vehículos podrían llevar instalado el sistema inteligente de decisión. Esto no sería una buena idea, porque habría mucha información que se omitiría y, por consiguiente, las decisiones tomadas no serían óptimas.

## **CAPÍTULO 10: CONCLUSIONES**

En este capítulo se realiza una conclusión sobre todo lo desarrollado, que ha dado lugar al proyecto actual. Se definen los parámetros que han caracterizado al mismo, y también se hace una reflexión crítica sobre los beneficios aportados y las limitaciones que lleva asociadas.

Uno de los aspectos más importantes a la hora de resolver situaciones de emergencia en la carretera es la predicción. Es primordial que se reaccione con antelación a que la situación crítica se suceda para poder tener una mejor respuesta por ambas partes, tanto el vehículo prioritario como el resto de vehículos que se encuentran en ese momento en la carretera, los cuales tienen que actuar de la manera apropiada para dejar paso libre al vehículo en emergencia. Todo este proceso debe ser lo más simple posible, e intentando no realizar demasiadas modificaciones al flujo normal del tráfico.

Como se comenta en [12], los beneficios obtenidos mediante la antelación sobre la situación de emergencia son principalmente los dos siguientes:

- ✓ Tiempos más rápidos de los vehículos prioritarios en llegar a sus destinos.
- ✓ Evitación de situaciones peligrosas en el camino del vehículo en emergencia, las cuales podrían crear nuevas dificultades e incluso nuevas situaciones de emergencia.

Las soluciones aportadas por el sistema inteligente desarrollado en el presente proyecto son solamente sugerencias; el sentido común de los conductores debe ser el factor más importante, el cual debería determinar la decisión final.

El trabajo desarrollado en el proyecto hace hincapié en diferentes áreas con respecto a los demás proyectos ya realizados. Acciones más detalladas son sugeridas a cada vehículo individualmente; los otros trabajos se centran más en el vehículo en situación de emergencia y en el estado global del tráfico para así poder ofrecer rutas alternativas al conductor de cada vehículo si hay atascos o tráfico lento en su camino.

Se pone especial atención en el proyecto a la distribución del tráfico una vez el vehículo en emergencia está cercano a los vehículos que se verán afectados. Sin embargo, en los demás trabajos todo se focaliza en momentos anteriores a la llegada del vehículo con prioridad hacia una concreta calle o lugar.

En la mayoría de los casos la información es recolectada en un nodo central para que éste procese todos los datos y tome una serie de decisiones que se transmitirán más tarde a los vehículos afectados por la situación. En este proyecto las situaciones de emergencia se tratan de diferente manera: se propone la integración de una ontología en cada vehículo para así dotarlos con capacidad de razonamiento y evitar el uso de un punto central.

Este sistema beneficia la organización del tráfico en situaciones de emergencia con decisiones inmediatas en las que cada vehículo es un punto de decisión, reacciona considerando sus vehículos adyacentes, y colabora con ellos para así llegar a un consenso en tiempo real. Las pruebas realizadas muestran tiempos de inferencia de menos de 300 milisegundos.

## ***BENEFICIOS DEL PROYECTO Y SUS CARACTERÍSTICAS***

El trabajo que ha sido desarrollado contiene gran cantidad de beneficios, algunos de ellos se explican a continuación:

- ✓ Se aporta una implementación seria, lo cual todavía no se ha hecho por ninguno de los estudios y proyectos realizados hasta el momento.

- ✓ Un simulador se ha construido, el cual representa el escenario que se estudia en el proyecto, para así poder realizar pruebas y comprobar cómo funcionaría el sistema real, instalado en los vehículos físicos.
- ✓ Se aporta una solución, lo cual es el punto fuerte del proyecto y lo que le diferencia de todos los demás, los cuales no ofrecen ninguna solución o las que aportan son demasiado básicas y generales.
- ✓ Se resuelven momentos en los que los conductores tienen dudas sobre cómo actuar como respuesta a una situación crítica en la carretera, relacionada con la seguridad. Cuando se resuelven estas situaciones, el factor más importante de generación de peligros prácticamente se anula (la indecisión o la duda).
- ✓ Las sugerencias comunicadas a los conductores son instrucciones simples y muy precisas. Así, la posibilidad de malinterpretar la información aportada y actuar de manera incorrecta se reduce drásticamente.
- ✓ El problema se resuelve mediante el estudio de la situación individual de cada vehículo por separado, independientemente de los demás que se encuentran en la carretera. De esta manera, una sugerencia que se informe a un vehículo concreto es de utilidad solamente para éste, y no se puede hacer extensible a ningún otro.

## ***LIMITACIONES DEL PROYECTO Y SUS CARACTERÍSTICAS***

A pesar de la gran cantidad de beneficios obtenidos, el trabajo realizado no es perfecto, existen algunas limitaciones que hay que tener en cuenta:

- El proyecto se basa en una idea fundamental: la innovación.  
Por lo tanto, el área sobre la que se trabaja es bastante difícil y la expectativa de éxito es baja, debido a la tasa de fracaso de los proyectos realizados en esta área.  
  
Sin embargo, como contrapunto a esta limitación, se ha realizado un artículo basado en el trabajo realizado en este proyecto que ha sido aprobado para una conferencia internacional en Calabria, Italia ('6th International Symposium on Intelligent Distributed Computing - IDC 2012'), lo cual aporta un éxito reconocido al proyecto desarrollado. (<http://idc2012.deis.unical.it/>)
- Cuando se dan instrucciones tan detalladas a los conductores, puede existir el peligro de que se cree una dependencia de los mismos hacia el dispositivo, anulando su capacidad de tomar decisiones por sí solos de alguna manera.  
Esta situación es difícil que se suceda porque las instrucciones comunicadas a los conductores no están presentes durante la conducción normal, sino exclusivamente en situaciones extraordinarias.
- Los conductores podrían estar en contra de este tipo de sistemas, ya que podría ser que no les gustase que les estuviesen continuamente sugiriendo acciones.

- Quizás las sugerencias dadas a los conductores pudieran llevarles a situaciones de confusión. Entonces, el sentido común de los mismos debería ser lo que opte por la mejor solución al problema.
- Podría ser posible que se requiriera formación básica a los conductores sobre cómo interpretar estas situaciones explicadas y las sugerencias recibidas por parte del dispositivo inteligente.  
Esta información podría ser impartida a los futuros conductores en el momento del aprendizaje para la obtención del permiso de conducir, en las academias de conducción.

Todo lo desarrollado ha sido minuciosamente estudiado para así intentar minimizar las limitaciones explicadas anteriormente. Éstas son ampliamente contrarrestadas con la gran cantidad y la calidad de los beneficios obtenidos que se ofrecen con el trabajo realizado.



## **CAPÍTULO 11: LÍNEAS FUTURAS**

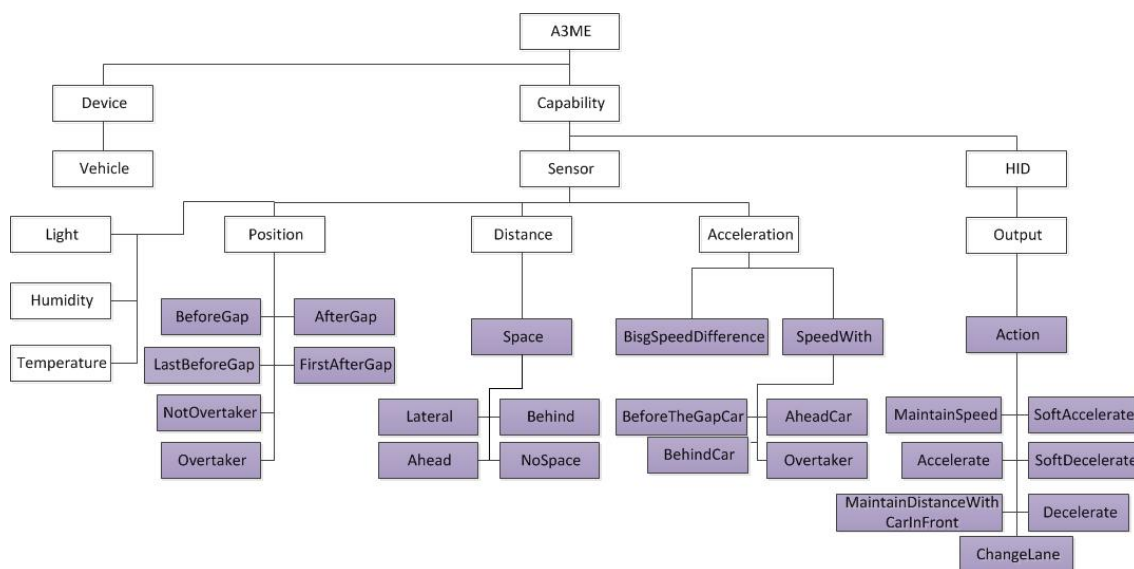
En este capítulo se presentan los siguientes pasos que se seguirán en el desarrollo del proyecto, los cuales aportarán una mayor calidad y precisión al trabajo desarrollado.

El siguiente paso a seguir en el desarrollo es la anexión de nuestra ontología a una ya estandarizada, para así dotar la nuestra de una base contrastada y con garantías de calidad a partir de la cual se puede extender la funcionalidad que ofrece.

Se ha elegido la ontología A3ME, ‘una clasificación básica y simple para la auto-descripción y el descubrimiento de dispositivos y sus capacidades en redes heterogéneas, incluyendo nodos de sensores de recursos limitados’. (<http://www.w3.org/2005/Incubator/ssn/XGR-ssn-20110628/#A3ME>)

Para su utilización, se sugiere que la misma se establezca como núcleo y se vaya construyendo una versión sofisticada semánticamente extendida de la ontología sobre este núcleo, lo que se ha hecho en el proyecto realizado. Esta extensión se ha realizado mediante la adición de clases para así modelar cada vehículo con respecto a sus adyacentes. Estas clases extendidas han sido: *position*, *distance* y *acceleration*. También se ha extendido la clase *output* para modelar las sugerencias que aporta la ontología como soluciones: *Accelerate*, *SoftAccelerate*, *Decelerate*, *SoftDecelerate*, *MaintainSpeed*, *MaintainDistanceWithCarInFront* y *ChangeLane*.

Todavía no se han modificado ni la ontología actual ni el simulador para adaptarse a la nueva estructura, lo cual no supondría un gran trabajo, pero sí se ha realizado ya el esquema de cómo quedarían las dos unidas. Éste se muestra en la siguiente figura:



**Ilustración 27: Ontología resultante a partir de la extensión de la ontología A3ME [16].**

En la Ilustración 27, los rectángulos de color violeta representan las clases que se han añadido a la ontología A3ME. Adicionalmente, se han integrados también otras clases que nos servirán para ampliarla de manera que se tenga en cuenta también la influencia de los fenómenos meteorológicos. Así, las decisiones tomadas serán mucho más completas y se adaptarán con más fidelidad a las situaciones reales.

También se pretende, en futuros pasos, la inclusión de escenarios más completos, como la ampliación del número de carriles por los que circulan los vehículos, tener en cuenta ambos sentidos de circulación del tráfico (no solamente uno, como hasta el momento) o cruces. Además, se pretende la integración de las infraestructuras situadas a lo largo de la carretera, como señales de tráfico, semáforos o puntos de información, lo cual supondrá una mayor cobertura del escenario, con lo que se conseguirá un control más

completo de la situación de emergencia y, por lo tanto, unas reacciones más eficientes por parte de los conductores.

Todo esto supondrá, como se pretende desde la idea principal del proyecto, mantener la seguridad en las situaciones de emergencia en la carretera, en las cuales está en peligro la vida de las personas.

# LOGROS CONSEGUIDOS

Con la realización del proyecto que se presenta en este documento se han conseguido una serie de logros que cabe destacar:

- ✓ Realización de un artículo técnico ('Short Paper'), llamado 'Ontology Based Road Traffic Management', para una conferencia internacional que se realizará en Calabria, Italia, en septiembre del 2012.  
'6th International Symposium on Intelligent Distributed Computing - IDC 2012'.  
(<http://idc2012.deis.unical.it/>)  
Esta conferencia trata sobre sistemas inteligentes distribuidos, y se acudirá a la misma con el objetivo de presentar el proyecto realizado.
- ✓ También se ha realizado una presentación en el CEIN, Centro Europeo de Empresas e Innovación de Navarra, en la cual se ha expuesto el trabajo realizado y se ha informado sobre la utilización de las ontologías a pequeñas empresas, las cuales buscan utilizar nuevas tecnologías en sus trabajos de innovación. (<http://www.cein.es/web/es/index.php>)

Finalmente, se está preparando otro artículo técnico, en el cual están puestas muchas expectativas, que será presentado a IEEE, el Instituto de Ingenierías Eléctricas y Electrónicas, que es 'la mayor asociación del mundo dedicada a la innovación tecnológica avanzada y a la excelencia para el beneficio de la humanidad'. (<http://www.ieee.org/about/index.html>)

# AGRADECIMIENTOS

Este proyecto ha sido realizado gracias a la ayuda del Gobierno de España, mediante una Beca de Tecnólogos.

También, es de agradecer la ayuda aportada por Jesús Villadangos, tutor del proyecto que se presenta en este documento, y de la beca sobre la que se basa el mismo. Él ha guiado todos y cada uno de los pasos que se han seguido para la consecución del proyecto realizado.

# REFERENCIAS

- [1] R. Baldessari, et al, Manifesto.  
URL [http://www.car-2-car.org/fileadmin/downloads/C2C-CC manifesto v1.1.pdf](http://www.car-2-car.org/fileadmin/downloads/C2C-CC_manifesto_v1.1.pdf)
- [2] S. A. Huss, A. Jaeger, H. Stuebing, simTD: Safe and Intelligent Mobility Field Test Germany; Architecture and Applications for Car-to-Car Communication, in: 48th ACM/EDAC/IEEE Design Automation Conference (DAC 2011): Workshop on Intra and Inter-Vehicle Networking: Past, Present, and Future, San Diego, 2011.
- [3] Fleetnet, <http://www.fleetnet.co.uk/>.
- [4] SEVECOM, Secure vehicular communication, <http://www.sevecom.org/>.
- [5] C. Gerken, NetWorks on wheels (1998 (accessed 02.11.2010)).  
URL <http://www.mindspring.com/~gerken/vehicles/>
- [6] Coopers, <http://www.coopers-ip.eu/index.php?id=project>.
- [7] Ist-Anemone, <http://www.cvisproject.org/en/links/anemone.htm>.
- [8] University Carlos III Madrid, Poseidon Project, <http://enjambre.it.uc3m.es/~poseidon/index.html>.
- [9] B. K. Mohandas, R. Liscano, O. W. W. Yang, Vehicle traffic congestion management in vehicular ad-hoc networks, in: LCN, IEEE, 2009.
- [10] S. R. Rizvi, S. Olariu, M. C. Weigle, M. E. Rizvi, A novel approach to reduce traffic chaos in emergency and evacuation scenarios, in: VTC Fall, IEEE, 2007.
- [11] Y. Yu, B. Liu, T. Wu, M. Liu, Flow-based Travel Plan via VANET, in: International Journal of Digital Content Technology and its Applications 5 (2011) 9.
- [12] A. Buchenscheit, F. Schaub, F. Kargl, M. Weber, A VANET-based Emergency Vehicle Warning System, in: First IEEE Vehicular Networking Conference (VNC 2009), IEEE, Tokyo, Japan, 2009.
- [13] G. H. Mohimani, F. Ashtiani, A. Javanmard, M. Hamdi, Mobility modeling, spatial traffic distribution, and probability of connectivity for sparse and dense vehicular ad hoc networks, IEEE Transactions on Vehicular Technology 58 (4) (2009) 1998–2007.
- [14] A. Ghazy, T. Ozkul, Design and simulation of an artificially intelligent VANET for solving traffic congestion, 6th International Symposium on Mechatronics and its Applications (2009) 1–6.
- [15] Sandia National Laboratories, Jess, the rule engine for the Java platform, Website <http://www.jessrules.com/> (Sep, 2009).
- [16] A. Herzog, D. Jacobi, A. Buchmann, A3me - an agent-based middleware approach for mixed mode environments, in: Proceedings of the 2008 The Second International Conference on Mobile Ubiquitous Computing, Systems, Services and Technologies, UBICOMM '08, IEEE Computer Society, Washington, DC, USA, 2008.
- [17] E. I. Europe, Gst rescue: Improving the response time of emergency services.
- [18] P. Papadimitratos, A. de la Fortelle, K. Evenssen, R. Brignolo, S. Cosenza, Vehicular Communication Systems: Enabling Technologies, Applications, and Future Outlook on Intelligent Transportation, IEEE Communications Magazine 47 (11) (2009) 84–95.

# APÉNDICE A

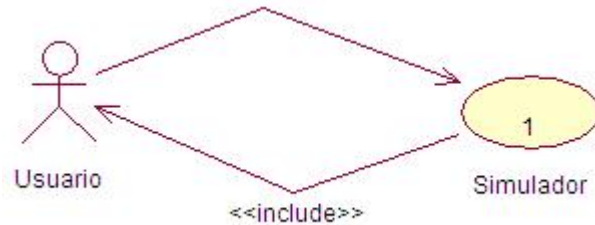
Gráfico que muestra la jerarquía de clases de la ontología:



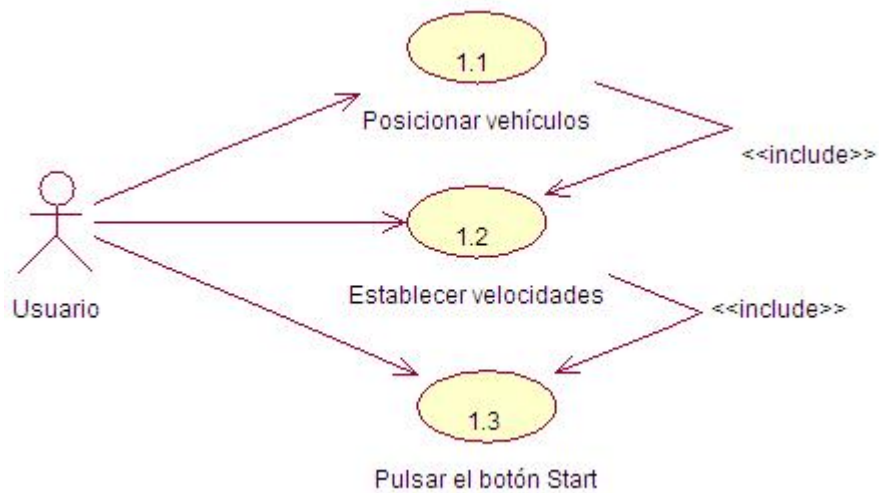
Ilustración 28: Jerarquía de clases de la ontología.

## APÉNDICE B

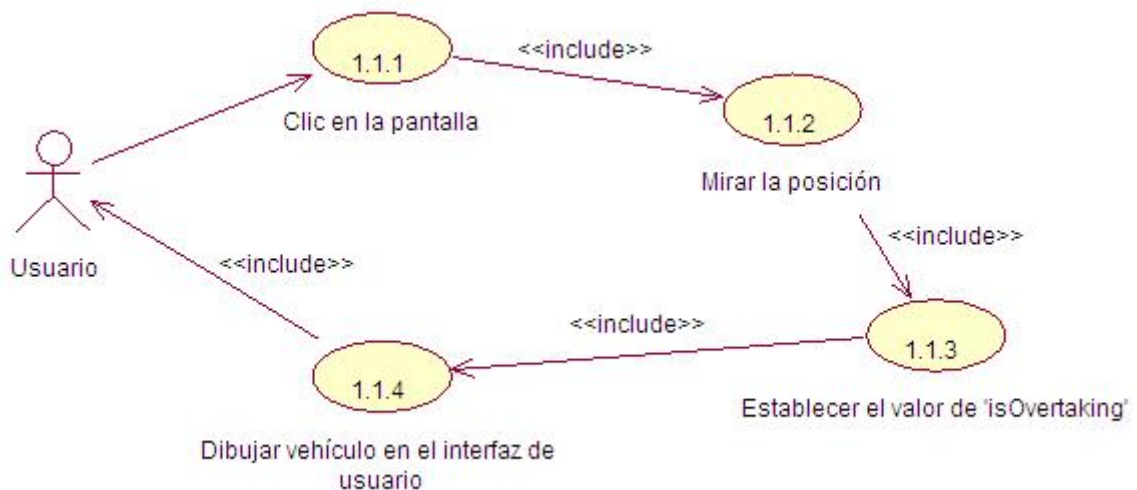
A continuación se muestran los casos de uso que muestran las funcionalidades requeridas a la aplicación desarrollada que actúa como simulador:



**Ilustración 29: Nivel 0, Traffic.**

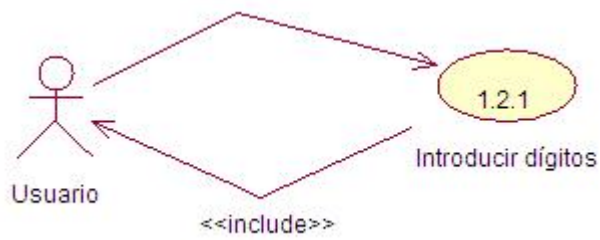


**Ilustración 30: Nivel 1, Simulador.**



**Ilustración 31: Nivel 1.1, Posicionar vehículo.**

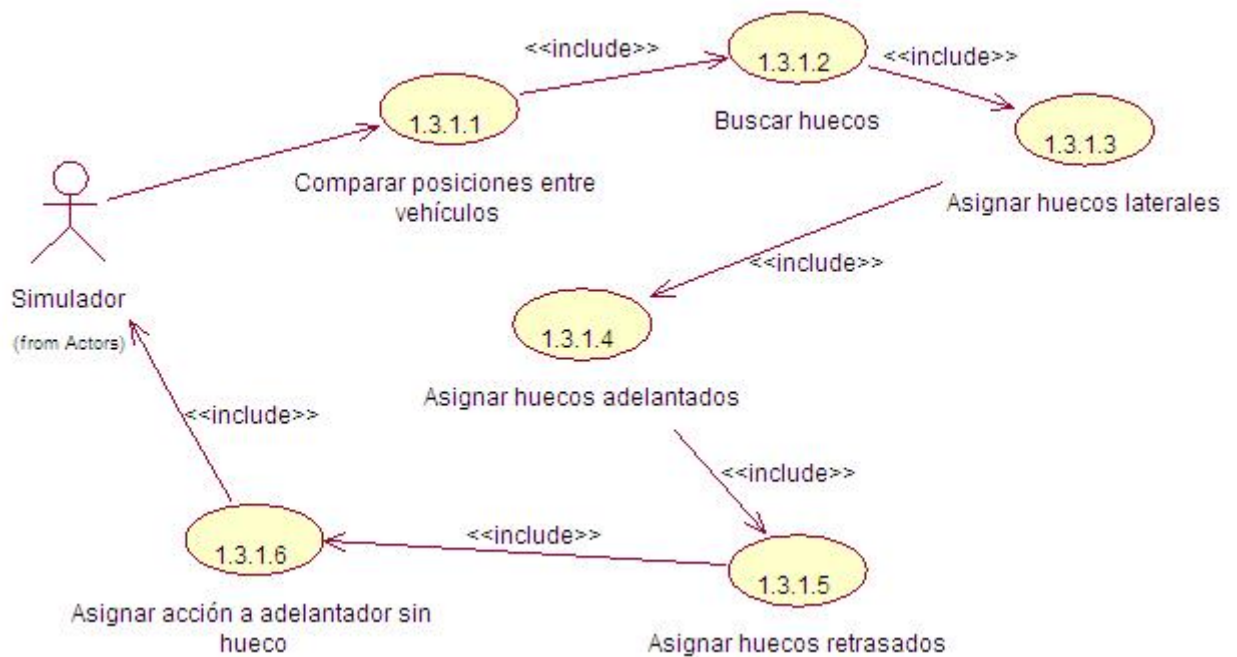




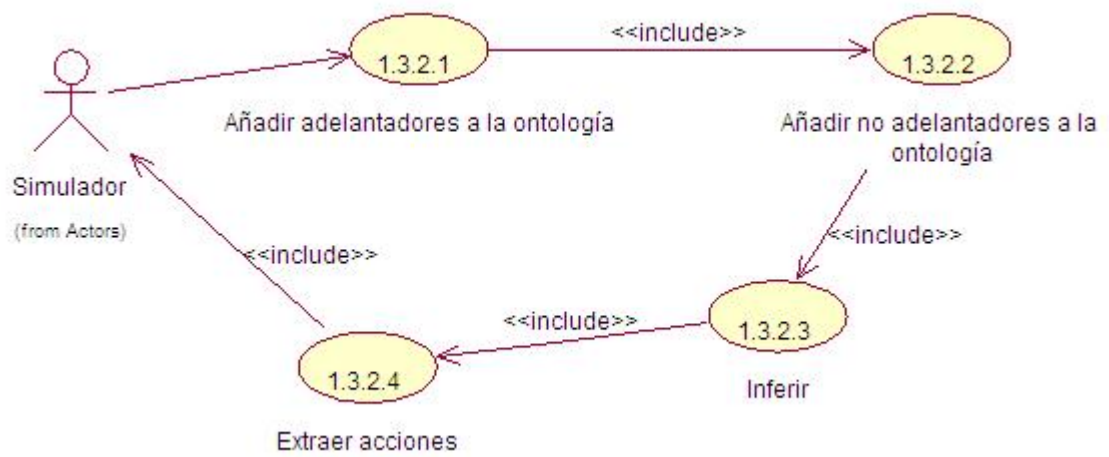
**Ilustración 32: Nivel 1.2, Establecer velocidad.**



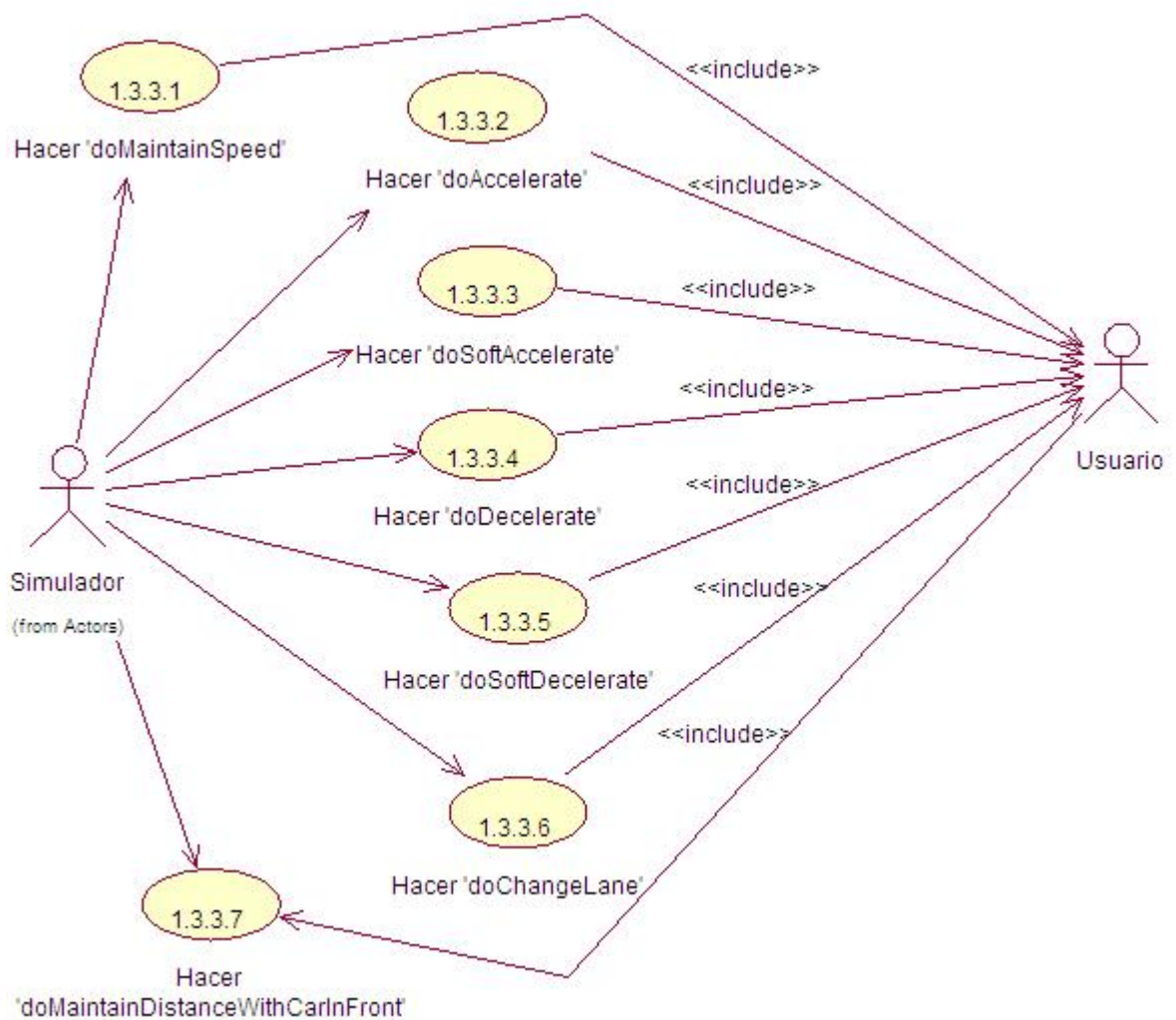
**Ilustración 33: Nivel 1.3, Start.**



**Ilustración 34: Nivel 1.3.1, No más vehículos.**



**Ilustración 35: Nivel 1.3.2, Utilizar ontología.**



**Ilustración 36: Nivel 1.3.3, Mover vehículos.**

## APÉNDICE C

Diagramas de colaboración, relacionados con los diagramas de secuencia mostrados en el apartado ‘Simulador’ del Capítulo 4, ‘Análisis’.

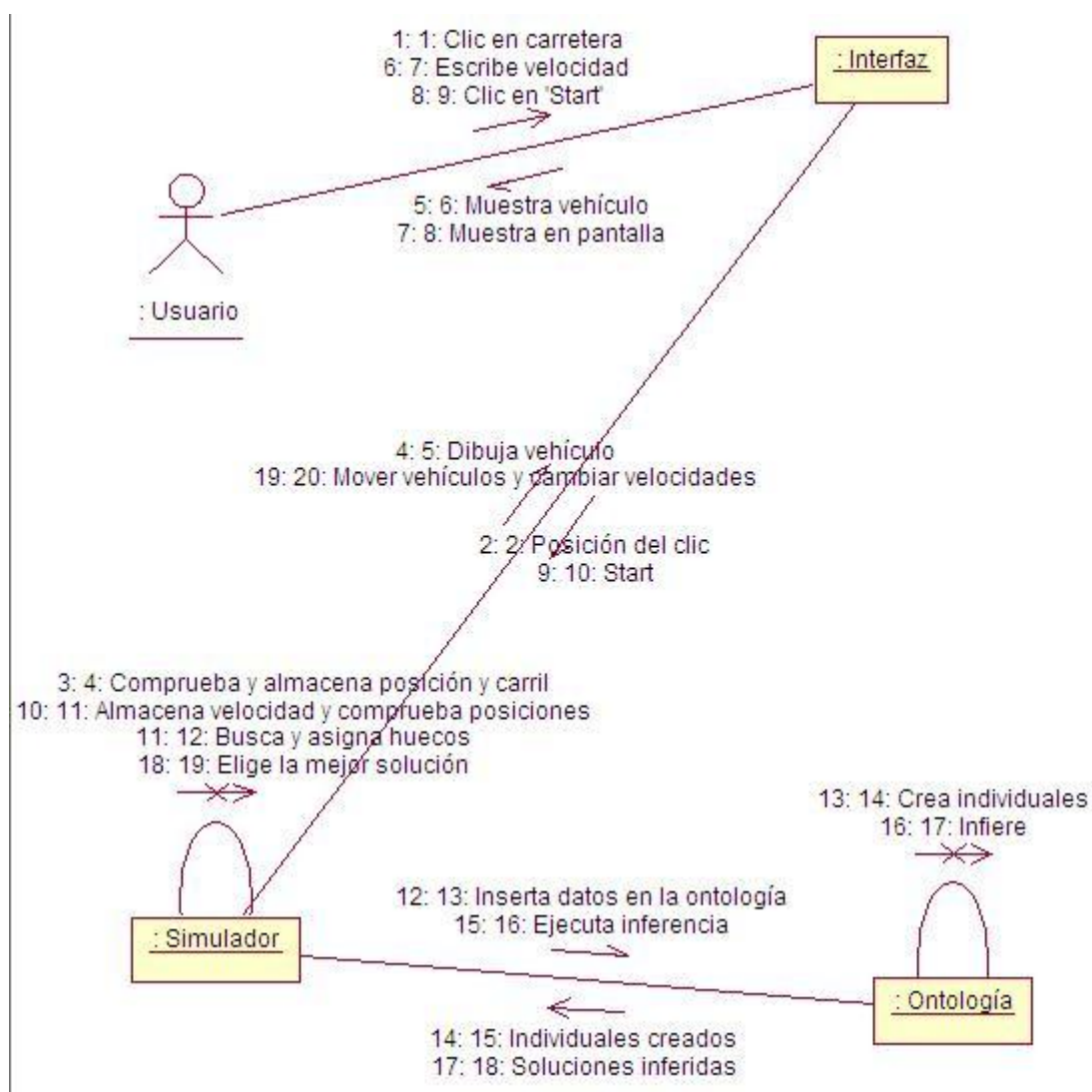


Ilustración 37: Diagrama de Colaboración - Simulador.





## APÉNDICE D

Las reglas SWRL incluidas en la ontología son:

[illegible]

✓	Rule-62	→	Overtaker(?c) ∧ Behind(?c) ∧ speedWithAheadCar(?c, 3) ∧ bigSpeedDifference(?c, false) ∧ speedWithBehindCar(?c, 2) → SoftDecelerate(?c)
✓	Rule-63	→	Overtaker(?c) ∧ NoSpace(?c) ∧ speedWithAheadCar(?c, 3) ∧ bigSpeedDifference(?c, false) ∧ speedWithBehindCar(?c, 2) → SoftDecelerate(?c)
✓	Rule-64	→	Overtaker(?c) ∧ Behind(?c) ∧ speedWithAheadCar(?c, 3) ∧ bigSpeedDifference(?c, false) ∧ speedWithBehindCar(?c, 1) → SoftDecelerate(?c)
✓	Rule-65	→	Overtaker(?c) ∧ NoSpace(?c) ∧ speedWithAheadCar(?c, 3) ∧ bigSpeedDifference(?c, false) ∧ speedWithBehindCar(?c, 1) → SoftDecelerate(?c)
✓	Rule-66	→	NotOvertaker(?c) ∧ Lateral(?c) ∧ BeforeGap(?c) ∧ speedWithBeforeTheGapCar(?c, 3) ∧ bigSpeedDifference(?c, false) → SoftDecelerate(?c)
✓	Rule-67	→	NotOvertaker(?c) ∧ Ahead(?c) ∧ BeforeGap(?c) ∧ speedWithBeforeTheGapCar(?c, 3) ∧ bigSpeedDifference(?c, false) → SoftDecelerate(?c)
✓	Rule-68	→	NotOvertaker(?c) ∧ Behind(?c) ∧ BeforeGap(?c) ∧ speedWithOvertaker(?c, 3) ∧ bigSpeedDifference(?c, false) → SoftDecelerate(?c)
✓	Rule-69	→	NotOvertaker(?c) ∧ Behind(?c) ∧ LastBeforeGap(?c) ∧ speedWithOvertaker(?c, 3) ∧ bigSpeedDifference(?c, false) → SoftDecelerate(?c)
✓	Rule-7	→	NotOvertaker(?c) ∧ Lateral(?c) ∧ FirstAfterGap(?c) ∧ speedWithBeforeTheGapCar(?c, 1) ∧ bigSpeedDifference(?c, true) → Accelerate(?c)
✓	Rule-70	→	NotOvertaker(?c) ∧ Behind(?c) ∧ BeforeGap(?c) ∧ speedWithOvertaker(?c, 2) → SoftDecelerate(?c)
✓	Rule-71	→	NotOvertaker(?c) ∧ Behind(?c) ∧ LastBeforeGap(?c) ∧ speedWithOvertaker(?c, 2) → SoftDecelerate(?c)
✓	Rule-72	→	NotOvertaker(?c) ∧ Behind(?c) ∧ AfterGap(?c) ∧ isInFrontOfOvertaker(?c, false) ∧ speedWithBeforeTheGapCar(?c, 2) → SoftDecelerate(?c)
✓	Rule-73	→	NotOvertaker(?c) ∧ Behind(?c) ∧ AfterGap(?c) ∧ isInFrontOfOvertaker(?c, false) ∧ speedWithBeforeTheGapCar(?c, 1) → SoftDecelerate(?c)
✓	Rule-74	→	NotOvertaker(?c) ∧ NoSpace(?c) ∧ isInFrontOfOvertaker(?c, false) → SoftDecelerate(?c)
✓	Rule-75	→	Vehicle(?c) ∧ isOvertaking(?c, false) → NotOvertaker(?c)
✓	Rule-76	→	Vehicle(?c) ∧ isOvertaking(?c, true) → Overtaker(?c)
✓	Rule-77	→	hasSpaceLateral(?c, true) → Lateral(?c)
✓	Rule-78	→	hasSpaceAhead(?c, true) ∧ hasSpaceLateral(?c, false) → Ahead(?c)
✓	Rule-79	→	hasSpaceAhead(?c, false) ∧ hasSpaceBehind(?c, true) ∧ hasSpaceLateral(?c, false) → Behind(?c)
✓	Rule-8	→	NotOvertaker(?c) ∧ Ahead(?c) ∧ FirstAfterGap(?c) ∧ speedWithBeforeTheGapCar(?c, 1) ∧ bigSpeedDifference(?c, true) → Accelerate(?c)
✓	Rule-80	→	hasSpaceAhead(?c, false) ∧ hasSpaceBehind(?c, false) ∧ hasSpaceLateral(?c, false) → NoSpace(?c)
✓	Rule-81	→	beforeTheGap(?c, true) ∧ lastFirst-Before-After-TheGap(?c, false) → BeforeGap(?c)
✓	Rule-82	→	beforeTheGap(?c, false) ∧ lastFirst-Before-After-TheGap(?c, false) → AfterGap(?c)
✓	Rule-83	→	beforeTheGap(?c, true) ∧ lastFirst-Before-After-TheGap(?c, true) → LastBeforeGap(?c)
✓	Rule-84	→	beforeTheGap(?c, false) ∧ lastFirst-Before-After-TheGap(?c, true) → FirstAfterGap(?c)
✓	Rule-85	→	Lateral(?c) ∧ Overtaker(?c) → ChangeLine(?c)
✓	Rule-9	→	NotOvertaker(?c) ∧ Lateral(?c) ∧ AfterGap(?c) ∧ speedWithBeforeTheGapCar(?c, 1) ∧ bigSpeedDifference(?c, true) → Accelerate(?c)

**Ilustración 39: Reglas SWRL.**

# APÉNDICE E

Las reglas incluidas en la ontología se basan en los siguientes patrones, que se extraen de los escenarios explicados en el apartado 2 de la Implementación:

## MaintainDistanceWithCarInFront

isOvertaking = F  $\rightarrow$  hasSpaceLateral = T  $\rightarrow$  beforeTheGap = T  $\rightarrow$  lastFirst-BeforeAfter-TheGap = T  
 $\rightarrow$  speedWithBeforeTheGapCar = "same"  $\rightarrow$  bigSpeedDifference = T or F  
 OR  
 TheGap = T  $\rightarrow$  hasSpaceLateral = F  $\rightarrow$  hasSpaceAhead = T  $\rightarrow$  beforeTheGap = T  $\rightarrow$  lastFirst-BeforeAfter-TheGap = T  
 $\rightarrow$  speedWithBeforeTheGapCar = "same"  $\rightarrow$  bigSpeedDifference = T or F

## ChangeLine

isOvertaking = T  $\rightarrow$  hasSpaceLateral = T

## MaintainSpeed

isOvertaking = T  $\rightarrow$   $\left\{ \begin{array}{l} \text{hasSpaceLateral} = \text{F} \\ \text{hasSpaceAhead} = \text{F} \\ \text{hasSpaceBehind} = \text{T or F} \end{array} \right\} \rightarrow$  speedWithBehindCar = "faster"  $\rightarrow$  bigSpeedDifference = T or F  
 AND  
 $\rightarrow$  speedWithAheadCar = "slowlier" OR "same"  
 $\rightarrow$  bigSpeedDifference = T or F

OR  
 isOvertaking = F  $\rightarrow$   $\left\{ \begin{array}{l} (\text{hasSpaceLateral} = \text{T}) \\ \text{OR} \\ (\text{hasSpaceLateral} = \text{F} \\ \text{hasSpaceAhead} = \text{T}) \end{array} \right\} \rightarrow$  beforeTheGap = T  $\rightarrow$  lastFirst-BeforeAfter-TheGap = F  
 $\rightarrow$  speedWithBeforeTheGapCar = "same" OR "slowlier"  
 $\rightarrow$  bigSpeedDifference = T or F  
 OR  
 $\rightarrow$  beforeTheGap = F  $\rightarrow$  lastFirst-BeforeAfter-TheGap = T or F  
 $\rightarrow$  speedWithBeforeTheGapCar = "same"  
 OR  
 $\rightarrow$  speedWithBeforeTheGapCar = "faster"  $\rightarrow$  bigSpeedDifference = T or F

OR

$\rightarrow$   $\left\{ \begin{array}{l} \text{hasSpaceLateral} = \text{F} \\ \text{hasSpaceAhead} = \text{F} \\ \text{hasSpaceBehind} = \text{T} \end{array} \right\} \rightarrow$  beforeTheGap = T  $\rightarrow$  lastFirst-BeforeAfter-TheGap = T or F  
 $\rightarrow$  speedWithOvertaker = "slowlier"  $\rightarrow$  bigSpeedDifference = T or F  
 OR  
 $\rightarrow$  beforeTheGap = F  $\rightarrow$  lastFirst-BeforeAfter-TheGap = F  
 $\rightarrow$  isInFrontOfOvertaker = T  
 $\rightarrow$  speedWithBehindCar = "faster"  $\rightarrow$  bigSpeedDifference = T or F



## Accelerate / SoftAccelerate

isOvertaking = T  $\rightarrow \left\{ \begin{array}{l} \text{hasSpaceLateral} = \text{F} \\ \text{hasSpaceAhead} = \text{T} \end{array} \right\} \rightarrow \text{speedWithBeforeTheGapCar} = \text{"same"} \rightarrow \text{bigSpeedDifference} = \text{T or F}$   
**ACCELERATE**

OR

$\rightarrow \text{speedWithBeforeTheGapCar} = \text{"slowlier"} \rightarrow \text{bigSpeedDifference} = \text{T or F}$   
**ACCELERATE**

OR

$\rightarrow \text{speedWithBeforeTheGapCar} = \text{"faster"} \rightarrow \text{bigSpeedDifference} = \text{T or F}$   
**SOFTACCELERATE**

OR

$\rightarrow \left\{ \begin{array}{l} \text{hasSpaceLateral} = \text{F} \\ \text{hasSpaceAhead} = \text{F} \\ \text{hasSpaceBehind} = \text{T or F} \end{array} \right\} \rightarrow \text{speedWithAheadCar} = \text{"same"} \text{ or } \text{"slowlier"} \\ \rightarrow \text{speedWithBehindCar} = \text{"same"} \rightarrow \text{bigSpeedDifference} = \text{T or F}$   
**SOFTACCELERATE**

OR

$\rightarrow \text{speedWithBehindCar} = \text{"slowlier"} \rightarrow \text{bigSpeedDifference} = \text{T}$   
**ACCELERATE**  
 $\rightarrow \text{bigSpeedDifference} = \text{F}$   
**SOFTACCELERATE**

OR

isOvertaking = F  $\rightarrow \left\{ \begin{array}{l} (\text{hasSpaceLateral} = \text{T}) \\ \text{OR} \\ (\text{hasSpaceLateral} = \text{F} \\ \text{hasSpaceAhead} = \text{T}) \end{array} \right\} \rightarrow \text{beforeTheGap} = \text{F} \rightarrow \text{lastFirst-BeforeAfter-TheGap} = \text{T or F}$   
 $\rightarrow \text{speedWithBeforeTheGapCar} = \text{"slowlier"} \rightarrow \text{bigSpeedDifference} = \text{T}$   
**ACCELERATE**

OR

$\rightarrow \text{bigSpeedDifference} = \text{F}$   
**SOFTACCELERATE**

OR

$\rightarrow \left\{ \begin{array}{l} \text{hasSpaceLateral} = \text{F} \\ \text{hasSpaceAhead} = \text{F} \\ \text{hasSpaceBehind} = \text{T} \end{array} \right\} \rightarrow \text{beforeTheGap} = \text{F} \rightarrow \text{lastFirst-BeforeAfter-TheGap} = \text{F}$   
 $\rightarrow \text{isInFrontOfOvertaker} = \text{T}$   
 $\rightarrow \text{speedWithBehindCar} = \text{"same"} \rightarrow \text{bigSpeedDifference} = \text{T or F}$   
**SOFTACCELERATE**

OR

$\rightarrow \text{speedWithBehindCar} = \text{"slowlier"} \rightarrow \text{bigSpeedDifference} = \text{T}$   
**ACCELERATE**

OR

$\rightarrow \text{bigSpeedDifference} = \text{F}$   
**SOFTACCELERATE**

OR

$\rightarrow \left\{ \begin{array}{l} \text{hasSpaceLateral} = \text{F} \\ \text{hasSpaceAhead} = \text{F} \\ \text{hasSpaceBehind} = \text{F} \end{array} \right\} \rightarrow \text{isInFrontOfOvertaker} = \text{T}$   
**SOFTACCELERATE**



## Decelerate / SoftDecelerate

isOvertaking = T  $\rightarrow \left\{ \begin{array}{l} \text{hasSpaceLateral} = \text{F} \\ \text{hasSpaceAhead} = \text{F} \\ \text{hasSpaceBehind} = \text{T or F} \end{array} \right\} \rightarrow \begin{array}{l} \text{speedWithBehindCar} = \text{"faster"} \text{ or "same"} \text{ or "slowlier"} \\ \rightarrow \text{speedWithAheadCar} = \text{"faster"} \rightarrow \text{bigSpeedDifference} = \text{T} \\ \textbf{DECELERATE} \end{array}$

OR  
 $\rightarrow \text{bigSpeedDifference} = \text{F}$   
**SOFTDECELERATE**

OR  
isOvertaking = F  $\rightarrow \left\{ \begin{array}{l} (\text{hasSpaceLateral} = \text{T}) \\ \text{OR} \\ (\text{hasSpaceLateral} = \text{F} \\ \text{hasSpaceAhead} = \text{T}) \end{array} \right\} \rightarrow \begin{array}{l} \text{beforeTheGap} = \text{T} \rightarrow \text{last/First-Before/After-The-Gap} = \text{F} \\ \rightarrow \text{speedWithBeforeTheGapCar} = \text{"faster"} \rightarrow \text{bigSpeedDifference} = \text{T} \\ \textbf{DECELERATE} \end{array}$

OR  
 $\rightarrow \text{bigSpeedDifference} = \text{F}$   
**SOFTDECELERATE**

OR  
 $\rightarrow \left\{ \begin{array}{l} \text{hasSpaceLateral} = \text{F} \\ \text{hasSpaceAhead} = \text{F} \\ \text{hasSpaceBehind} = \text{T} \end{array} \right\} \rightarrow \begin{array}{l} \text{beforeTheGap} = \text{T} \rightarrow \text{lastFirst-BeforeAfter-TheGap} = \text{T or F} \\ \rightarrow \text{speedWithOvertaker} = \text{"same"} \rightarrow \text{bigSpeedDifference} = \text{T or F} \\ \textbf{SOFTDECELERATE} \end{array}$

OR  
 $\rightarrow \text{speedWithOvertaker} = \text{"faster"} \rightarrow \text{bigSpeedDifference} = \text{T}$   
**DECELERATE**  
 $\rightarrow \text{bigSpeedDifference} = \text{F}$   
**SOFTDECELERATE**

OR  
 $\rightarrow \text{beforeTheGap} = \text{F} \rightarrow \text{lastFirst-BeforeAfter-TheGap} = \text{F}$   
 $\rightarrow \text{isInFrontOfOvertaker} = \text{F}$   
 $\rightarrow \text{speedWithBeforeTheGapCar} = \text{"same"} \rightarrow \text{bigSpeedDifference} = \text{T or F}$   
**SOFTDECELERATE**

OR  
 $\rightarrow \text{speedWithBeforeTheGapCar} = \text{"faster"} \rightarrow \text{bigSpeedDifference} = \text{T}$   
**DECELERATE**

OR  
 $\rightarrow \text{bigSpeedDifference} = \text{F}$   
**SOFTDECELERATE**

OR  
 $\rightarrow \left\{ \begin{array}{l} \text{hasSpaceLateral} = \text{F} \\ \text{hasSpaceAhead} = \text{F} \\ \text{hasSpaceBehind} = \text{F} \end{array} \right\} \rightarrow \begin{array}{l} \text{isInFrontOfOvertaker} = \text{F} \\ \textbf{SOFTDECELERATE} \end{array}$

